


43-15061

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION  
CONTRACT NO. NASW-6

Technical Report No. 32-67

**CODING THEORY AND ITS APPLICATIONS  
TO COMMUNICATIONS SYSTEMS**

Leonard Baumert  
Mahlon Easterling  
Solomon W. Golomb  
Andrew Viterbi



---

W. K. Victor, *Chief*  
*Communications Systems Research Section*

JET PROPULSION LABORATORY  
CALIFORNIA INSTITUTE OF TECHNOLOGY  
PASADENA, CALIFORNIA  
March 31, 1961

Copyright © 1961  
Jet Propulsion Laboratory  
California Institute of Technology

## FIGURES

1.	Waveforms and sequences .....	20
2.	Correlation computation .....	21
3.	Correlation function of PN code .....	24
4.	Calculation of the correlation function of two PN sequences .....	25
5.	Calculation of the correlation function of two PN sequences by probabilities .....	26
6.	Calculation of the autocorrelation function of a combination sequence .....	27
7.	Simplified calculation of the autocorrelation function of a combination sequence .....	30
8.	Autocorrelation function of a combination which has no extra peaks .....	31
9.	Calculation of the correlation function of a combination with one of its components .....	32
10.	Correlation detector for a noisy signal .....	32
11.	Method of generating an error signal for tracking a code .....	34
12.	Double-loop code tracking device .....	34
13.	Computation of error function for double-loop high-frequency clock .....	36
14.	Computation of error function for double-loop best clock frequency .....	37
15.	Calculation of autocorrelation function for ranging sequences .....	38
16.	System for acquiring a three-component code .....	39
17.	Correlation of a combination sequence with its components .....	40
18.	Example of alternating PN and clock .....	42
19.	Generator for maximal-length linear shift register sequence .....	43
20.	Phases of maximal-length linear shift register sequence .....	44
21.	State diagram for a sequence generator .....	47
22.	Basic communications-system model .....	50
23.	Binary coded phase-coherent system for transmission of 2 bits/word .....	52
24.	Autocorrelation function of shift register code .....	58
25.	Shift register and generated code .....	58

## CONTENTS

I.	Introduction .....	1
II.	Codes with Special Correlation .....	3
A.	Simplex Codes .....	4
B.	Orthogonal and Biorthogonal Codes .....	6
C.	Sequences with Two-Level Autocorrelation .....	7
D.	Hadamard Matrices .....	9
E.	Hadamard Designs .....	11
F.	Difference Sets .....	13
G.	Other Constraints on Codes .....	16
III.	A Pseudorandom Coded Ranging System .....	18
A.	Correlation Properties of Codes .....	19
B.	Correlation and Tracking Schemes .....	32
C.	Acquirable Codes .....	36
D.	Generation and Manipulation of Sequences .....	43
IV.	Coded Phase-Coherent Communications .....	49
A.	The Basic Model .....	49
B.	Realization of the Model .....	51
C.	Binary Codes .....	56
D.	Optimal Decision and Probability of Error .....	60
E.	Bit Error Probabilities .....	66
F.	Information Rate and Channel Capacity .....	72
G.	Conclusions .....	78
	References .....	82

## FIGURES (Cont'd)

26.	Word error probability—orthogonal codes .....	62
27.	Word error probability—biorthogonal codes .....	65
28.	Word error probability—uncoded .....	67
29.	Comparison of coded and uncoded word error probabilities; $n = 5$ .....	68
30.	Comparison of coded and uncoded word error probabilities; $n = 10$ .....	69
31.	Bit error probability—orthogonal codes .....	70
32.	Bit error probability—biorthogonal codes .....	72
33.	Diagram of transition probabilities in the presence of noise—orthogonal codes .....	73
34.	Received information rate—orthogonal codes .....	74
35.	Channel capacity—orthogonal codes .....	76
36.	Channel efficiency—orthogonal codes .....	77
37.	Diagram of transition probabilities in the presence of noise—biorthogonal codes .....	77
38.	Received information rate—biorthogonal codes .....	79
39.	Channel capacity—biorthogonal codes .....	80
40.	Channel efficiency—biorthogonal codes .....	81

## ABSTRACT

15061

A general theory of binary sequences with desirable correlation properties is developed for application to the design of digital communication systems. The underlying mathematical problems are the existence, construction, and properties of "orthogonal matrices" or, as they are also known, "Hadamard designs." The first detailed application is to the design of a ranging system with unambiguous, high-precision resolution over interplanetary distances which can nonetheless be quickly synchronized and yield its range data in real time. The second major application is to the design and analysis of optimum digital telemetry systems. It is shown that for several categories of codes with suitable orthogonality properties, the theoretical bound on the information rate is actually approached, as the number of code words increases, for specific input signal-to-noise conditions.

## I. INTRODUCTION

Historically, the basic mathematical tool in radio communication theory has been Fourier time-and-frequency analysis. The RF signal is generally regarded as a linear combination of sine waves; and the classical concept of modulation involves the variation of one of the three parameters (amplitude, frequency, or phase) associated with a pure sine wave so as to carry information.

In recent years there has been increasing emphasis on so-called *digital communications*. For purposes of this report, the digital signal may be regarded conceptually as a sequence of *ones* and *zeros* or of *ones* and *minus ones*. In actual practice, there could be either a *pulse train* in which *one* is a *pulse* and *zero* is a *no-pulse*, or a high-frequency sine wave (called a *continuous wave* or *CW signal*) for which *one* is a phase shift of  $+90^\circ$  and *minus one* is a phase shift of  $-90^\circ$ , each lasting unit duration. From the classical standpoint, the pulse/no-pulse sequence is an amplitude-modulated square wave, while the  $+90^\circ/-90^\circ$  sequence is a phase-modulated sine wave.

In changing the emphasis from the study of sine waves to the study of binary sequences (i.e., sequences of *ones* and *zeros* or of *ones* and *minus ones*), certain facts have stood out. One of the important properties of sine waves is that all the harmonics  $\sin nx$  of the fundamental  $\sin x$  are mutually orthogonal on the standard interval  $(0, 2\pi)$ . Also,  $\sin x$  is orthogonal to two of its phase shifts,  $\cos x$  and  $-\cos x$ . It has been found that orthogonality properties of this sort are among the most desirable attributes of signals in a wide variety of communications situations.

In statistical terms, orthogonal means *uncorrelated*. Whenever one has a set of possible messages to encode for a communication link, one would like their encoded forms to be as mutually distinct as possible. This is approximately achieved in the *orthogonal* or *uncorrelated* case. If one has two or more messages to encode, it will be seen that it is actually possible to achieve mutual negative correlation among them; but as the number of messages increases, the negative correlation coefficients tend to zero.

Section II of this report is devoted to the existence and construction of orthogonal and transorthogonal sets of code words. As is so often the case with discrete problems of this sort, there are simple upper bounds to the parameters in question; but owing to combinatorial limitations, the attainment of these bounds does not occur in all cases.

In special cases, the cyclic phase shifts of a single sequence form an approximately orthogonal collection of code words. This is of particular importance for applications to range radar, where the amount of time displacement,

or phase shift, between the transmitted signal and the returning reflected signal is directly proportional to range. Here the possible received messages are the phase shifts of the transmitted signal; and the ideal state of affairs is for the various phase shifts to be as mutually distinguishable as possible. The problems of existence and construction for such sequences are included in Section II, while the applications to ranging are treated in Section III. One of the most interesting features of the ranging systems under discussion is the possibility of establishing synchronization between transmitted and received signals in a very short span of time, even over interplanetary ranges.

For telemetry applications, the only requirement is that the code words be as mutually distinct as possible. They need not be phase shifts of one another. In Section IV of this report, orthogonal and biorthogonal telemetry codes will be evaluated from the standpoint of information theory. In the limit, as the number of code words increases, the theoretical bound on the information rate of a noisy channel is actually attained for suitable input signal-to-noise conditions.

## II. CODES WITH SPECIAL CORRELATION

In this section the discussion is restricted to *uniform binary* codes. Thus, all the code words will contain the same number of symbols, and these symbols will be chosen from a two-letter alphabet. Furthermore, the alphabet will usually be 1, -1. For these purposes, then, a *code* is a collection of  $n$  vectors from a  $w$ -dimensional vector space. The vectors are referred to as *code words* having  $w$  symbols per word;  $w$  is the *word length* of the code.

The *correlation*  $C(x, y)$  of two  $w$ -dimensional vectors  $x, y$  is given by

$$C(x, y) = \frac{1}{w} \sum_{i=1}^w x_i y_i$$

If  $x, y$  are vectors of *ones* and *minus ones*, then  $C(x, y)$  is the cosine of the angle between them. Two vectors  $x, y$  are said to be *orthogonal* if  $C(x, y) = 0$ .

The *autocorrelation function*  $C_x(j)$  of a  $w$ -dimensional vector  $x$  is given by

$$C_x(j) = \frac{1}{w} \sum_{i=1}^w x_i x_{i+j} \quad \text{where } x_{w+k} = x_k \text{ by definition}$$

For example, if  $x = (1, -1, -1, 1)$  then

$$C_x(3) = \frac{1}{4} \sum_{i=1}^w x_i x_{i+3} = \frac{1}{4} [1 \cdot 1 + (-1) \cdot 1 + (-1) \cdot (-1) + 1 \cdot (-1)] = \frac{1}{4} [0] = 0$$

and

$$C_x(0) = 1, C_x(1) = 0, C_x(2) = -1$$

In communications applications it is often desirable to use binary vectors which do not consist of *ones* and *minus ones*. Usually it is desirable to preserve the correlation properties present in the 1, -1 representation. This can be done by defining correlation more generally. Thus if

$$C(x, y) = \frac{A - D}{A + D}$$

where  $A$  is the number of agreements of  $x$  with  $y$  and  $D$  is the number of disagreements, a "generalized correlation" has been defined, which includes the one previously discussed.

### A. Simplex Codes

A code is called *maximally transorthogonal* if it consists of  $n$  optimally distinguishable code words; that is, if the correlation between distinct code words is minimized. This minimum is always negative, hence the word transorthogonal. It is important to note that this definition does not fix or limit the word length in any way. A bound on the transorthogonality of binary codes is given by the following theorem. Letting  $v_i$ ,  $i = 1, 2, \dots, n$ , be the code words, and 1, -1 be the symbols, then:

*Theorem.*

$$\min_{\text{all codes}} \max_{i \neq j} C(v_i, v_j) \geq \begin{cases} \frac{-1}{n-1} & \text{if } n \text{ is even} \\ \frac{-1}{n} & \text{if } n \text{ is odd} \end{cases}$$

*Proof.* Consider the  $n \times w$  matrix  $M$  whose  $i$ th row is  $v_i$ . Then  $1/w (MM^T)$  is the symmetric matrix of correlation coefficients. The average correlation  $\bar{c}$  of  $v_i, v_j (i \neq j)$  is given by

$$\begin{aligned} \bar{c} &= \frac{1}{n(n-1)} \sum_{i \neq j} C(v_i, v_j) = \frac{1}{n(n-1)} \sum_{i,j} C(v_i, v_j) - \sum_i C(v_i, v_i) \\ &= \frac{1}{n(n-1)w} \left( \sum_{i,j} v_i \cdot v_j - \sum_i v_i \cdot v_i \right) = \frac{1}{n(n-1)w} \left( \left| \sum_{i=1}^n v_i \right|^2 - \sum_i |v_i|^2 \right) \end{aligned}$$

but  $|v_i|^2 = w$  for all  $i$ , and letting  $V = \sum_{i=1}^n v_i$ , one sees that  $V$  is the sum of the row vectors of  $M$ . Thus  $|V|^2$  is the sum of the squares of the column sums. Hence, in order to minimize  $\bar{c}$ , one must minimize  $|V|^2$ . For even  $n$ , all the column sums can be made equal to zero; but for odd  $n$ , one can do no better than 1 or -1. Thus

$$\min_{\text{all codes}} \bar{c} = \min_{\text{all codes}} \frac{1}{n(n-1)w} (|V|^2 - nw) \geq \begin{cases} \frac{1}{n(n-1)w} (-nw) = \frac{-1}{n-1} & \text{if } n \text{ is even} \\ \frac{1}{n(n-1)w} (w - nw) = -\frac{1}{n} & \text{if } n \text{ is odd} \end{cases}$$

But

$$\min_{\text{all codes}} \max_{i \neq j} C(v_i, v_j) \geq \min_{\text{all codes}} \text{average}_{i \neq j} C(v_i, v_j) = \min_{\text{all codes}} \bar{c}$$

which completes the proof.

In the course of this proof it has also been shown that:

*Corollary.*

$$\min_{\text{all codes}} \text{average}_{i \neq j} C(v_i, v_j) \geq \begin{cases} \frac{-1}{n-1} & \text{if } n \text{ is even} \\ \frac{-1}{n} & \text{if } n \text{ is odd} \end{cases}$$

A code achieving this bound on its maximum correlation  $v_i, v_j (i \neq j)$  is called a *simplex* code. Simplex codes exist for an infinite number of values of  $n$ . In particular, all  $n \leq 100$  have associated simplex codes, except possibly  $n = 45, 46, 57, 58, 77, 78, 91, 92, 93, 94$ . The existence or nonexistence of simplex codes in general is tied up with the mathematical theory of Hadamard matrices. Since an introduction to this subject is provided in a later section, a pair of examples of simplex codes will suffice for the present.

For  $n = 7$  there is the code

$$\begin{array}{ccccccc} -1 & 1 & 1 & -1 & 1 & -1 & -1 \\ 1 & 1 & -1 & 1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & -1 & 1 \\ -1 & 1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & -1 & 1 & 1 & -1 \\ -1 & -1 & -1 & 1 & 1 & -1 & 1 \\ -1 & -1 & 1 & 1 & -1 & 1 & -1 \end{array}$$

while  $n = 14$  gives

1	-1	-1	-1	-1	1	1	1	1	-1	1	1	1	-1	1	1	-1	-1	-1	1	-1	1	-1	-1	1	-1
-1	1	-1	-1	-1	1	1	1	1	1	-1	-1	1	1	-1	1	1	1	-1	-1	-1	-1	1	-1	-1	1
-1	-1	1	1	1	-1	1	-1	1	1	-1	1	1	-1	1	-1	1	-1	1	-1	-1	1	-1	-1	-1	1
-1	-1	1	1	1	-1	-1	1	-1	1	1	-1	1	1	1	1	-1	-1	-1	1	-1	-1	1	1	-1	-1
-1	-1	1	1	1	1	-1	-1	1	-1	1	1	-1	1	-1	1	1	1	-1	-1	1	-1	-1	-1	1	-1
-1	1	-1	1	-1	-1	1	-1	-1	1	-1	-1	-1	-1	1	1	1	1	-1	1	1	1	-1	1	1	-1
1	-1	1	-1	-1	1	-1	-1	1	-1	-1	-1	-1	-1	1	1	1	-1	1	1	1	-1	1	1	-1	1
1	-1	-1	-1	1	-1	1	-1	1	1	1	-1	1	-1	-1	-1	-1	1	1	-1	1	-1	1	1	1	-1
-1	1	1	-1	-1	-1	-1	1	1	1	1	-1	-1	1	-1	-1	-1	-1	1	1	1	1	-1	-1	1	1
1	-1	-1	1	-1	-1	-1	1	-1	-1	-1	1	1	1	-1	1	-1	1	1	-1	1	1	-1	1	-1	1
-1	1	-1	-1	1	1	-1	-1	-1	-1	-1	1	1	1	-1	-1	1	-1	1	1	-1	1	1	1	1	-1
1	1	1	-1	1	1	-1	1	-1	1	-1	1	-1	1	-1	1	1	-1	-1	-1	-1	-1	-1	1	1	1
1	1	1	1	-1	-1	1	1	-1	-1	1	1	-1	-1	-1	-1	1	1	1	1	-1	-1	1	-1	-1	-1
1	1	-1	1	1	1	1	-1	-1	-1	1	-1	-1	1	1	-1	-1	-1	-1	-1	1	1	1	-1	-1	1

Note that in the first example the word length satisfies  $w = n$ , whereas in the second example  $w = 2(n - 1)$ . It can be demonstrated fairly easily that these values of  $w$  are minimal. Specifically, it can be shown that if  $n \neq 1, 2$  and if  $n = 4t + 1$  or  $n = 4t + 2$ , then the word length is a multiple of  $2(4t + 1)$ . If  $n = 4t$  or  $4t - 1$ , however, then the word length is a multiple of  $4t - 1$ . Thus both examples illustrate minimum word length.

## B. Orthogonal and Biorthogonal Codes

For most applications of binary codes, the distinction between correlations of  $-1/(n - 1)$  and 0 is small enough to be disregarded without serious loss. Thus, instead of demanding optimum distinguishability (a simplex code), one may ask that  $C(v_i, v_j) = 0$  ( $i \neq j$ ). A code satisfying this requirement is called *orthogonal*. For orthogonal codes, it is usually assumed the  $n = w$ , and for this it is necessary that  $n = 1, 2$ , or  $4t$ , as will be shown in Section II-D.

A biorthogonal code consists of the vectors of an orthogonal code and their negatives. Thus  $n = 2w$ , in general, and  $w$  is limited to 1, 2, or  $4t$  as mentioned above. The correlation properties of biorthogonal codes are  $C(v_i, \pm v_j) = 0$  ( $i \neq j$ ),  $C(v_i, -v_i) = -1$ . Perhaps the best known biorthogonal codes are the first-order Reed-Muller codes, for which  $n = 2^k$  (Ref. 1). An example in 0,1 notation is

0 0 0 0 0 0 0 0	1 1 1 1 1 1 1 1
0 1 0 1 0 1 0 1	1 0 1 0 1 0 1 0
0 0 1 1 0 0 1 1	1 1 0 0 1 1 0 0
0 1 1 0 0 1 1 0	1 0 0 1 1 0 0 1
0 0 0 0 1 1 1 1	1 1 1 1 0 0 0 0
0 1 0 1 1 0 1 0	1 0 1 0 0 1 0 1
0 0 1 1 1 1 0 0	1 1 0 0 0 0 1 1
0 1 1 0 1 0 0 1	1 0 0 1 0 1 1 0

Another example of a biorthogonal code is the one containing the 24 code words:

0 0 0 0 0 0 0 0 0 0 0 0	1 1 1 1 1 1 1 1 1 1 1 1
0 1 1 0 1 1 1 0 0 0 1 0	1 0 0 1 0 0 0 1 1 1 0 1
0 1 0 1 1 1 0 0 0 1 0 1	1 0 1 0 0 0 1 1 1 0 1 0
0 0 1 1 1 0 0 0 1 0 1 1	1 1 0 0 0 1 1 1 0 1 0 0
0 1 1 1 0 0 0 1 0 1 1 0	1 0 0 0 1 1 1 0 1 0 0 1
0 1 1 0 0 0 1 0 1 1 0 1	1 0 0 1 1 1 0 1 0 0 1 0
0 1 0 0 0 1 0 1 1 0 1 1	1 0 1 1 1 0 1 0 0 1 0 0
0 0 0 0 1 0 1 1 0 1 1 1	1 1 1 1 0 1 0 0 1 0 0 0
0 0 0 1 0 1 1 0 1 1 1 0	1 1 1 0 1 0 0 1 0 0 0 1
0 0 1 0 1 1 0 1 1 1 0 0	1 1 0 1 0 0 1 0 0 0 1 1
0 1 0 1 1 0 1 1 1 0 0 0	1 0 1 0 0 1 0 0 0 1 1 1
0 0 1 1 0 1 1 1 0 0 0 1	1 1 0 0 1 0 0 0 1 1 1 0

### C. Sequences with Two-Level Autocorrelation

If the correlation function of a  $w$ -dimensional vector  $x$  is such that

$$C_x(0) = \frac{1}{w} \sum_{i=1}^w x_i^2 = 1$$

$$C_x(j) = \frac{1}{w} \sum_{i=1}^w x_i x_{i+j} = a \neq 1 \quad (1 \leq j < w)$$

then  $x$  (considered as a sequence) is said to have a *flat* or *two-level* autocorrelation function, or, briefly, to be a *two-level sequence*. If, moreover,  $C_x(j) = -1/w$ , the sequence is sometimes called a *pseudonoise* (PN) sequence, (Ref. 2), or, rather loosely, an *orthogonal sequence*.

An example of a two-level sequence is 1 1 1 -1, where the out-of-phase correlation is 0. Thus this sequence and its cyclic shifts can be taken as the code words of an orthogonal code. Another two-level sequence is -1 1 1 -1 1 -1 -1, the out-of-phase correlation here being  $-1/7$ . Thus one has a sequence whose *shifts* form a simplex code, indeed the first simplex code mentioned above.

An introduction to the theory and construction of two-level sequences is given in a later section (II-D) concerned with difference sets. Perhaps the best known and most exhaustively investigated sequences of this type are the *m-sequences* (also called *maximal length linear recurring sequences* or *maximal length linear shift register sequences* (Ref. 2, 3). These sequences are of length  $w = 2^k - 1$ , with  $C_x(j) = -1/2^k - 1$ ,  $1 \leq j < 2^k - 1$ .

There are four known types of two-level sequences giving rise to simplex codes:

- |  |  |
|--|--|
| (1) $w = 2^k - 1$                                  | ( <i>m-sequences</i> )                                   |
| (2) $w = 4t - 1$ is prime                          | ("quadratic residue" or "Legendre" sequences, Ref. 4, 5) |
| (3) $w = 4t - 1 = 4x^2 + 27$ is prime, $x > 0$     | (Hall sequences, Ref. 6)                                 |
| (4) $w = p(p + 2)$ where both $p, p + 2$ are prime | (twin prime sequences, Ref. 7)                           |

These four types of sequences overlap to some extent. Restricting attention to the sequence length  $w$ :

- (1) and (2) overlap if  $w$  is a Mersenne prime.
- (1) and (3) overlap if  $w = 31, 127, 131071$  (Ref. 8).
- (1) and (4) overlap if  $w = 15$ .
- (3) is a subset of (2).

The known Mersenne primes are  $w = 2^k - 1$  with  $k = 2, 3, 5, 7, 13, 17, 19, 31, 61, 89, 107, 127, 521, 607, 1279, 2203, 2281$ . Thus the first few lengths are  $w = 3, 7, 31, 127, 8191, 131071$ . In most cases these overlaps result in distinct sequences. Specifically, (2) and (3) always lead to distinct sequences. While there is some overlapping for smaller value of  $w$ ,  $w = 31$  is the first value of  $w$  leading to truly *distinct* sequences.

## D. Hadamard Matrices

An *Hadamard matrix* (Ref. 4, 7, 9, 10) is a square matrix whose elements are *ones* and *minus ones* and whose row vectors are mutually orthogonal (equivalently, whose column vectors are mutually orthogonal). For example,

$$(a) \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$(b) \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ \hline 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

$$(c) \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ \hline 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ \hline 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix}$$

$$(d) \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \hline 1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 \\ 1 & 1 & 1 & -1 & 1 & -1 & -1 & -1 \\ 1 & 1 & -1 & 1 & -1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 & -1 & 1 & 1 & -1 \\ 1 & -1 & -1 & -1 & 1 & 1 & -1 & 1 \\ 1 & -1 & -1 & 1 & 1 & -1 & 1 & -1 \end{bmatrix}$$

(The partitioning exhibits structural properties which will be discussed later.)

The determinant of an Hadamard matrix achieves the upper bound specified by Hadamard's "determinantal inequality" (Ref. 11), i.e.,

$$|H|^2 = \pi \sum_{i=1}^w \sum_{j=1}^w h_{ij}^2 = w^w$$

It is clear from the definition of these matrices that one may

- (1) interchange rows
- (2) interchange columns
- (3) change the sign of every element in a row
- (4) change the sign of every element in a column

without disturbing the Hadamard property. Using these operations it is possible to establish a *normal form* for

Hadamard matrices by insisting that the first column contain only *ones*. All examples given above are in normal form. If two Hadamard matrices can be transformed into each other by operations of the type listed above, they are called equivalent. For example, (c) and (d) above are equivalent. This example demonstrates that the normal form is not unique within an equivalence class.

The existence of Hadamard matrices of all possible dimensions is an unsolved problem in mathematics. A result of interest is:

*Theorem.* If  $m \geq 1$  is the dimension of a Hadamard matrix, then  $m = 1, 2$ , or  $4t$ .

*Proof.* [1] is clearly a  $1 \times 1$  Hadamard matrix, (a) above is  $2 \times 2$ . If a Hadamard matrix has at least 3 row vectors  $x, y, z$ , then

$$\sum_{i=1}^m (x_i + y_i)(x_i + z_i) = \sum_{i=1}^m (x_i^2 + x_i z_i + y_i x_i + y_i z_i) = \sum_{i=1}^m x_i^2 = m$$

but the summands on the left-hand side are all multiples of 4. Thus  $m = 4t$ .

Whereas explicit methods of construction have been given for an infinite number of  $m$ 's ( $=4t$ ), there are still an infinite number of unsolved cases. It has been conjectured that Hadamard matrices exist for all  $m = 4t$ , but this has neither been proved nor disproved. There is more encouragement, however, from an applications point of view. Explicit constructions have been given for all Hadamard matrices of order  $\leq 200$  with but 5 exceptions (Ref. 4, 10). These are  $m = 92, 116, 156, 184, 188$ . In this regard, perhaps the simplest and most powerful result is the following:

*Theorem.* If  $H_1$  and  $H_2$  are Hadamard matrices, then so is  $H_1 \times H_2$ , where  $H_1 \times H_2$  is formed by substituting  $H_2$  for 1 and  $-H_2$  for -1 in  $H_1$  (the "Kronecker product").

The proof is by straightforward verification and will not be included here. By way of example, however, (b) above is partitioned to show that it is  $(a) \times (a)$ , and (c) above is partitioned to show that it is  $(a) \times (b) = (a) \times [(a) \times (a)]$ .

The connection between orthogonal (and thus biorthogonal) codes and Hadamard matrices should be clear at this point. The following theorem demonstrates the connection between Hadamard matrices and simplex codes.

*Theorem.* If a Hadamard matrix exists for  $m = 4t$ , then simplex codes exist for  $m = 4t, 4t - 1, 2t$ , and  $2t - 1$ .

*Proof.* If the first column of an Hadamard matrix in normal form is deleted, the row vectors form a simplex code for  $m = 4t$ . Deleting any vector of this collection leads to a solution for  $m = 4t - 1$ . Since the columns of an Hadamard matrix are mutually orthogonal, any column other than the first of an Hadamard matrix in normal form contains  $2t$  ones and  $2t$  minus ones. Thus, taking an Hadamard matrix in normal form and selecting a noninitial column  $j$ , the row vectors whose entries in column  $j$  are one (or, alternatively, minus one) may be deleted. If the first and  $j$ th columns are also deleted, the partial row vectors remaining form a set of  $2t$  vectors of length  $4t - 2$  whose correlation with each other is  $-2/(4t - 2) = -1/(2t - 1)$ : that is, a simplex code for  $m = 2t$ . As before, one of these vectors may be deleted to give a solution for  $m = 2t - 1$ .

Another problem intimately connected with Hadamard matrices is that of symmetric balanced incomplete block designs (the so-called  $v, k, \lambda$  problem). It is discussed in the following section.

## E. Hadamard Designs

A *balanced incomplete block design* (Ref. 12) is an arrangement of  $v$  objects into  $b$  sets in such a manner that:

- (1) each set contains exactly  $k$  different objects.
- (2) each object occurs in exactly  $r$  different sets.
- (3) any pair of objects occurs in exactly  $\lambda$  different sets.

These parameters satisfy the following two relations:

- (1)  $bk = vr$ .
- (2)  $\lambda(v - 1) = r(k - 1)$ .

The second of these relations remains nontrivial even in the case of a symmetric design (where  $b = v$ , and, consequently,  $k = r$ ). Thus (2) becomes

$$\lambda(v - 1) = k(k - 1)$$

If a  $v \times v$  matrix  $D$  is formed of zeros and ones, letting the columns represent the  $v$  objects and the rows the  $b = v$  sets, and if one puts  $d_{ij} = 1$  when the  $j$ th object occurs in the  $i$ th set but  $d_{ij} = 0$  otherwise, then  $D$  is called an *incidence matrix* of the block design.

For example, let  $v = 7$ ,  $k = 3$ ,  $\lambda = 1$ . Then

$$\begin{array}{c}
 \begin{array}{cccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
 S_1 & \begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 0 & 0 \end{bmatrix} \\
 S_2 & \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \\
 S_3 & \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \\
 S_4 & \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \\
 S_5 & \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix} \\
 S_6 & \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix} \\
 S_7 & \begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix}
 \end{array}
 \end{array}
 \quad \text{and} \quad
 \begin{array}{c}
 \begin{array}{cccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
 S_1 & \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix} \\
 S_2 & \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix} \\
 S_3 & \begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix} \\
 S_4 & \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \\
 S_5 & \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 & 1 \end{bmatrix} \\
 S_6 & \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \\
 S_7 & \begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix}
 \end{array}
 \end{array}$$

are incidence matrices of two solutions to the design problem.

The careful reader may have noticed that, apart from superficial differences, these matrices correspond to the Hadamard matrices (d) and (c) respectively. Herein lies the connection between symmetric balanced incomplete block designs and Hadamard matrices. More specifically, there is the following theorem:

*Theorem.* An Hadamard matrix for  $m = 4t$  exists if and only if there exists a symmetric balanced incomplete block design with parameters  $v = 4t - 1$ ,  $k = 2t - 1$ ,  $\lambda = t - 1$  (usually called an Hadamard design).

*Proof.* Given an Hadamard matrix in normal form, delete the first row and column. Since the row (column) vectors of a Hadamard matrix are orthogonal, there must be  $2t - 1 = k$  ones in each remaining row (column). Next, correlate two column vectors of this reduced matrix. Letting  $1 \cdot 1$  occur  $\alpha$  times, it is seen that  $1 \cdot (-1)$  and  $(-1) \cdot 1$  occur  $(2t - 1 - \alpha)$  times each, and that  $(-1) \cdot (-1)$  occurs  $\alpha + 1$  times. Thus

$$(4t - 1) C(x, y) = 2\alpha + 1 - 2(2t - 1 - \alpha) = 4\alpha - 4t + 3 - 1$$

so that

$$\alpha = t - 1 = \lambda$$

Thus, identifying  $-1$  with  $0$  and  $1$  with  $1$ , it has been shown that the reduced Hadamard matrix is the incidence matrix of a Hadamard design. Conversely, given an Hadamard design, an incidence matrix may be formed using  $-1$  in place of  $0$ . Now, correlating column vectors  $x, y$ , there is the product  $1 \cdot 1$  occurring  $\lambda = t - 1$  times,  $1 \cdot (-1)$  and  $(-1) \cdot 1$  occurring  $k - \lambda = t$  times each, and  $(-1) \cdot (-1)$  occurring  $v - k - (k - \lambda) = t$  times, so that

$$(4t - 1) C(x, y) = 2t - 1 - 2t = -1$$

Thus the addition of a row and column of ones to the matrix would create mutual orthogonality of the column vectors, i.e., an Hadamard matrix.

By definition, the incidence matrix exhibited by an Hadamard matrix in normal form is the *core* of the Hadamard matrix. In Hadamard matrix (d) above, the core is partitioned off from the rest of the matrix. This matrix exhibits another phenomenon not previously mentioned, namely, the fact that the row vectors of its core consist of all cyclic shifts of the first vector. An Hadamard design having such an incidence matrix is called *cyclic*. Cyclic designs are discussed further in the following section.

## F. Difference Sets

A *difference set*  $D = \{d_1, d_2, \dots, d_k\}$  is a subset of the integers modulo  $v$  for which

$$d_i - d_j \pmod{v} \quad (i \neq j)$$

assumes each value  $1, 2, \dots, v-1$  exactly  $\lambda$  times (Ref. 6, 13-15).

For example, if  $v = 7$ ,  $k = 3$ ,  $\lambda = 1$ , then  $D = \{1, 2, 4\}$  is a difference set with these parameters. Specifically,

$$\left. \begin{array}{l} d_1 - d_2 = 1 - 2 = -1 \equiv 6 \\ d_1 - d_3 = 1 - 4 = -3 \equiv 4 \\ d_2 - d_3 = 2 - 4 = -2 \equiv 5 \\ d_2 - d_1 = 2 - 1 = 1 \\ d_3 - d_1 = 4 - 1 = 3 \\ d_3 - d_2 = 4 - 2 = 2 \end{array} \right\} \quad (\text{all congruences modulo } 7)$$

As might be suspected from the occurrence of the parameters  $v$ ,  $k$ ,  $\lambda$ , there is a direct connection between difference sets and block designs. In particular, there is the following correspondence.

*Theorem.* A difference set exists for particular values of the parameters  $v$ ,  $k$ ,  $\lambda$  if and only if a cyclic symmetric balanced incomplete block design exists for the same values of  $v$ ,  $k$ ,  $\lambda$ .

*Proof.* Suppose a difference set  $D = \{d_1, d_2, \dots, d_k\}$  exists with parameters  $v$ ,  $k$ ,  $\lambda$ . Considering  $0, 1, 2, \dots, v-1$  as the objects, let

$$S_{i+1} = \{d_{1+i}, d_{2+i}, \dots, d_{k+i}\} \quad 0 \leq i \leq v-1$$

be the sets of the design, where the subscripts on the  $d$ 's are computed modulo  $v$ . Clearly, the resulting  $v$ -sets have the following properties:

- (1) Each set contains exactly  $k$  different objects.
- (2) Each object occurs in exactly  $k$  different sets.
- (3)  $b = v$ ,  $k = r$  (symmetric).
- (4) The sets are all of the cyclic shifts of  $D$  modulo  $v$  (cyclic).

Further, using (4), and since  $d_i - d_j \equiv m$  (modulo  $v$ ) has exactly  $\lambda$  solutions in  $D$ , any pair  $d_i, d_j$  occurs in exactly  $\lambda$  of the sets. Thus the sets  $S_i$  form a cyclic symmetric balanced incomplete block design.

Given a cyclic symmetric balanced incomplete block design consisting of sets  $S_i$ , let  $D$  be the set formed from  $S_1$  by putting  $i$  (modulo  $v$ ) in  $D$ , if the  $i$ th object occurs in  $S_1$ . If there are any solutions to  $i_j - i_q \equiv m \pmod{v}$  in  $D$  for a particular value of  $m \neq 0$ , there are exactly  $\lambda$  of them, since each pair of objects (in particular the  $i_j$ th and the  $i_q$ th) occurs exactly  $\lambda$  times in the cyclic design. But there are  $k(k-1)$  ordered pairs  $i_j, i_q$  ( $i_j \neq i_q$ ) in  $D$ , and thus there must be  $k(k-1)/\lambda$  values of  $m$  (modulo  $v$ ). But  $k(k-1) = \lambda(v-1)$  in all symmetric designs (see Section II-E), so all non-zero residues modulo  $v$  are represented exactly  $\lambda$  times in this form. That is,  $D$  is a difference set.

Given a difference set  $D$ , consider the vector  $v$  whose  $i$ th component is 1 if  $d_i$  is in  $D$  and  $-1$  otherwise. Correlating  $v$  with its cyclic shifts yields  $1 \cdot 1$  exactly  $\lambda$  times,  $1 \cdot (-1)$  and  $(-1) \cdot 1$ ,  $k - \lambda$  times each,  $(-1) \cdot (-1)$  occurs  $v - 2(k - \lambda) - \lambda$  times, and one obtains

$$C_v(j) = \frac{v - 4(k - \lambda)}{v} \quad (1 \leq j < v)$$

$$C_v(0) = 1$$

That is,  $v$  is a sequence with two-level autocorrelation. Conversely, given a sequence of *ones* and *minus ones* with two-level autocorrelation, there is an associated difference set. Thus, methods of constructing difference sets can be applied to the construction of two-level sequences.

As a step in this direction, it is helpful to introduce the following definition. If  $t$  is an integer such that in some order  $td_1, td_2, \dots, td_k$  are  $d_1 + s, d_2 + s, \dots, d_k + s \pmod{v}$ , then  $t$  is called a *multiplier* of the difference set  $D = \{d_1, d_2, \dots, d_k\}$ . In the terminology of modern algebra, a multiplier is an *automorphism* of the associated cyclic block design; and the multipliers form a group called the *multiplier group* of the design.

*Theorem.* Let  $n_1$  divide  $n(=k-\lambda)$ ,  $(n_1, v) = 1$  and  $n_1 > \lambda$ . If for every prime  $p$  dividing  $n_1$  there is a  $j$  such that

$$p^j \equiv t \pmod{v}$$

then  $t$  is a multiplier of a difference set modulo  $v$ . The proof will not be given here (Ref. 6). Instead, the usefulness of this theorem will be illustrated by constructing the difference set  $v = 23$ ,  $k = 11$ ,  $\lambda = 5$ . Let  $n_1 = 6$ ; then  $t \equiv 2^5 \equiv 3^2 \equiv 9 \pmod{23}$  is a multiplier. Suppose a difference set  $D$  exists. Then

$$\left. \begin{array}{l} 9D \equiv D + s \\ 9\{D+i\} \equiv D + s + 9i \\ i \equiv s + 9i \\ 9\{D+i\} \equiv D + i \\ 8i \equiv -s \end{array} \right\} \quad (\text{all mod } 23)$$

Thus if

there exists a shift  $i$  such that

But this means solving

which can be done since

$$(8, 23) = 1$$

Thus one may break the residues mod 23 up into sets which satisfy  $9\{D+i\} \equiv D+i \pmod{23}$ . Doing so yields 3 sets

$$\begin{aligned} \{1, 9, 12, 16, 6, 8, 3, 4, 13, 2, 18\} &\equiv \{\text{distinct powers of } 9 \pmod{23}\} \\ \{5, 7, 10, 11, 14, 15, 17, 19, 20, 21, 22\} \\ \{0\} \end{aligned}$$

each set belonging to  $D+i$  completely or not at all. Now, it is necessary to find  $k = 11$  residues which form a difference set. Clearly, one may choose only the first or second sets for this. For this example, either will do (although this is not generally the case). Thus

$$\{1, 2, 3, 4, 6, 8, 9, 12, 13, 16, 18\}$$

is a difference set for  $v = 23$ ,  $k = 11$ ,  $\lambda = 5$ .

The method here demonstrated can be used to construct difference sets when they exist, and also to prove nonexistence. It has been conjectured that any prime  $p$  satisfying  $(p, v) = 1$  and dividing  $n = k - \lambda$  is a multiplier. This is true for all known difference sets but has not been shown to be true in general.

#### G. Other Constraints on Codes

Suppose a Hadamard matrix exists for  $n_0 = 4t_0$ , and suppose further that it is required to form a simplex, orthogonal or biorthogonal code for  $n_0$ . There are a considerable number of Hadamard matrices to choose from in general. That is, whereas large numbers of these matrices are considered equivalent from a mathematical point of view, this is not necessarily the case from a coding point of view. Specifically, if there is some property which is considered desirable for a code to possess, and if that property is not invariant under permutations and complementations

$$\left\{ \begin{array}{l} (0 \leftrightarrow 1) \\ (1 \leftrightarrow -1) \end{array} \right\}$$

of rows and columns, one should expect that some equivalent Hadamard matrices possess this property to a larger degree than others. Naturally, one would like to generate the code from a matrix possessing this property to the largest degree.

There are many such properties which could arise, but this section will be limited to a discussion of only two of them in order to indicate the considerations involved. These typical side-conditions are:

- (1) Ease of code generation.
- (2) Self-synchronization (the use of "comma-free" codes).

Of all the Hadamard designs considered above, perhaps the simplest to generate are those which are cyclic. In this case, one need only generate or store one code word, and then use its shifts for the other words. Within this class of two-level sequences, the easiest to generate are the maximal-length linear shift register sequences. These have length  $2^n - 1$  and can be generated very easily with an  $n$ -stage shift register (Ref. 2, 3, 16, 17). Thus a ranking in order of ease-of-generation might be

- (1) Cyclic designs with word length  $2^n - 1$ .
- (2) Cyclic designs of other lengths.
- (3) Hadamard designs which are noncyclic.

With regard to self-synchronism, one can require that a code be "comma-free" (Ref. 18). A code  $C$  with uniform word length  $w$  is said to be *comma-free* if  $a_1, a_2, \dots, a_w$  and  $b_1, b_2, \dots, b_w$  in  $C$  imply  $a_2, \dots, a_w, b_1; a_3, \dots, b_1, b_2; \dots; a_w, b_1, \dots, b_{w-1}$  are not in  $C$  (i.e., no overlaps of code words are code words). It is clear that with a comma-free code it takes no more than  $2w - 2$  symbols to establish word synchronization, in the absence of noise. Note, moreover, that these  $2w - 2$  symbols need not be the first  $2w - 2$  transmitted. Thus the receiver would not have to get the first part of a message to establish word synchronism in the part it did pick up. Such a code is said to be *uniquely decipherable in the small*.

Comma-free orthogonal and biorthogonal codes have been found. One example of a comma-free orthogonal code is the following:

```

1 1 1 1 1 0 1 1
1 1 1 1 0 1 0 0
0 1 1 0 1 1 1 0
0 1 1 0 0 0 0 1
0 1 0 1 0 0 1 0
0 1 0 1 1 1 0 1
1 1 0 0 0 1 1 1
1 1 0 0 1 0 0 0

```

### III. A PSEUDORANDOM CODED RANGING SYSTEM

The purpose of this section is to describe a pseudorandom coded ranging system and to explain some of the coding techniques which have been worked out to implement such a system.

The classical radar technique for ranging is to transmit a pulse and measure the time until the return of the reflected pulse. The elapsed time multiplied by the propagation velocity in the medium is twice the range. As the range increases, it becomes increasingly difficult to detect the reflected pulse even if a transponder is used to enhance the echo. One may resort to transmitting many pulses or even a square wave or sine wave and apply correlation detection methods to detect the returned signal. However, this leads to an ambiguity in the range measurement if the repetition period of the pulses or the period of the square wave or sine wave is less than the time required for the signal to travel to the target and back. To resolve this ambiguity an additional periodicity must be imposed on the transmitted signal which is longer than the time required for the echo to return. Furthermore, this additional periodicity must be such that the phase of the returned signal can be resolved to within one pulse repetition period or one period of the square wave or sine wave.

PN sequences, described in the previous section, have just this property. If the sequence of the pulses and no pulses is transmitted where the pulses represent the zeros, then the returned signal can be correlated with a locally generated model to determine the exact phase of the returned signal. Since these sequences can be generated with periods in the billions and trillions, there is no problem in resolving the range ambiguity.

There is still one final problem before one starts to design a system. To achieve fine range resolution, one desires a high pulse repetition rate. For long ranges this implies a long sequence. Long ranges also imply a low returned signal level and therefore a long integration time to detect the signal. One can determine the phase of a PN sequence by correlation only by trial and error; that is, one chooses a phase and tries a correlation. If the phase chosen is not correct, there is no better choice than to try the next phase. If the sequence were a million digits long, all of the million possible phases might have to be tried in order to find the right one. However, from an information theory standpoint only  $\log_2 10^6$  yes-or-no questions should have to be asked instead of  $10^6$  questions. Sequences have been found in which the phase can be determined by correlation using fewer trials. These sequences are formed by combining several short PN sequences digit by digit. If the periods of the several sequences have no common divisors, the period of the combined sequence is the product of the periods of the several sequences. It is possible to determine the phase of the combined sequence by determining separately the phases of the component

sequences. This requires at most  $p_1 + p_2 + \dots + p_n$  trial correlations to determine the phase of a sequence whose length is  $p_1 p_2 \dots p_n$  when the  $p_i$  are the lengths of the component sequences.

## A. Correlation Properties of Codes

The subsequent material in this section is concerned with properties of these sequences and with some techniques which have been developed for system design.

1. *Sequences and waveforms.* If a signal is made up of equally spaced pulses, it is easy to impose the sequence properties on the signal by assigning a pulse to each digit and deleting those pulses which correspond to a zero. This technique was used in the *Venus* radar experiment conducted by the Massachusetts Institute of Technology. However, for correlation detection, particularly when a carrier is used, it is desirable to have a continuous carrier wave with phase modulation. Therefore, it is more suitable to use a waveform for each digit instead of a pulse. The waveform that has been used is a dc level for a specific period of time with a unit positive level for zeros and a unit negative level for ones. An example of a sequence and the corresponding waveform is shown in Fig. 1a. The product of two waveforms is shown in Fig. 1b, and the digit-by-digit modulo 2 sum of the corresponding sequences is shown in Fig. 1c. The table in Fig. 1d summarizes the relations between the waveform and the sequence. The waveform is referred to as a code. In this section, computations and algebraic representations will usually be in terms of sequences, but block diagrams and descriptions of mechanizations will be in terms of codes.

2. *Correlation functions of sequences and codes.* The correlation properties of PN sequences are the properties that are used in constructing acquirable codes. The (unnormalized) correlation between two sequences of the same length is defined as:

$$C = \text{number of agreements} - \text{number of disagreements}$$

The correlation between two sequences of different lengths  $p_1$  and  $p_2$  may be obtained by repeating the second sequence  $p_1$  times and the first sequence  $p_2$  times to obtain two sequences of length  $p_1 p_2$ .

The (cross-) correlation function of two sequences of the same length is the correlation of the one sequence with all of the cyclic permutations of the other sequence. If the two sequences are of different lengths, they are repeated as described above. The autocorrelation function of a sequence is the correlation of the sequence with all cyclic permutations of itself.

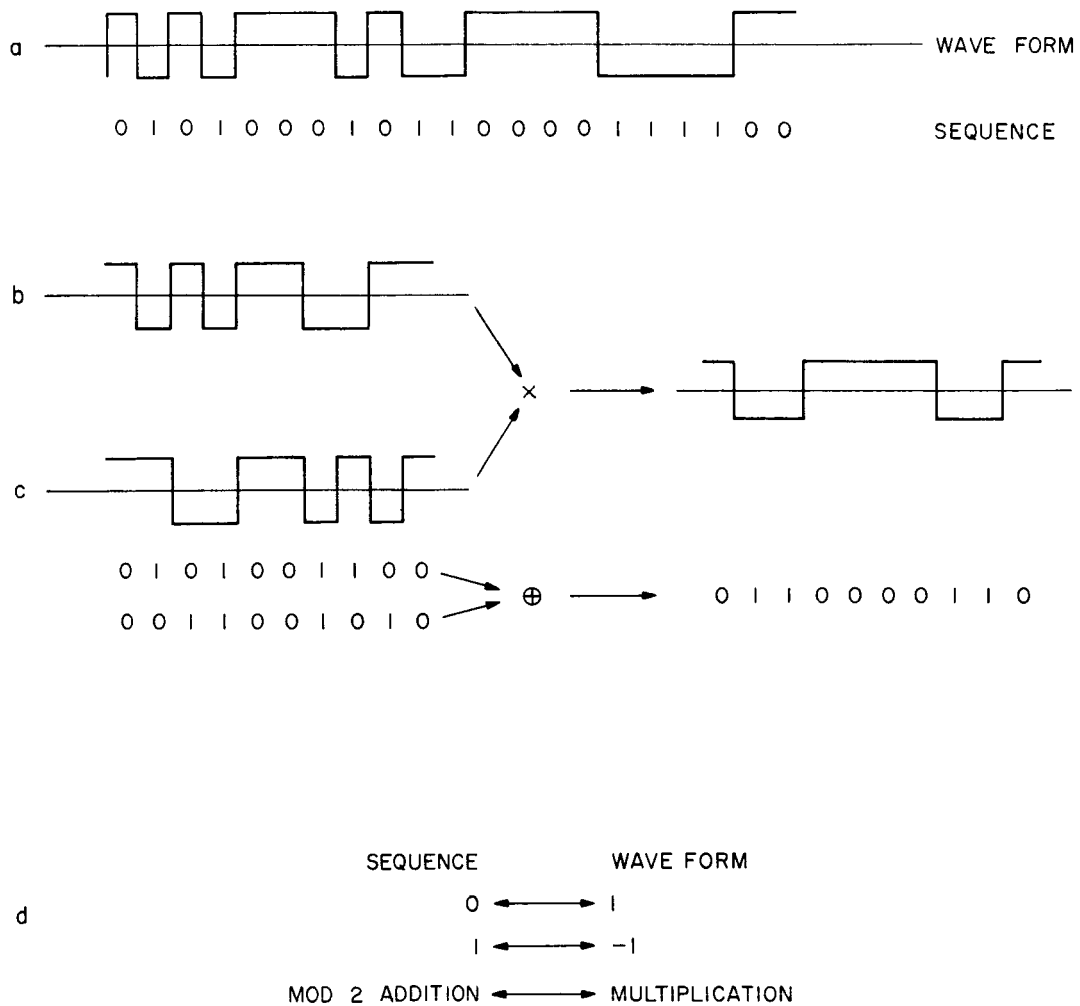


Fig. 1. Waveforms and sequences

It is often desirable to use the *normalized correlation* defined by

$$C_n = \frac{\text{number of agreements} - \text{number of disagreements}}{\text{number of agreements} + \text{number of disagreements}}$$

As in Section II, the *normalized correlation function* may also be used.

To compute the correlation of two sequences, it is convenient to compute the modulo 2 sum of the two sequences digit by digit and, noting that agreements lead to *zeros* and disagreements lead to *ones*, apply the following formula:

$$C = \text{number of zeros} - \text{number of ones}$$

An example of the computation of a correlation is given in Fig. 2a. An example of the computation of a correlation function is given in Fig. 2b. The new variable  $\tau$  is the number of digits by which the second sequence is cycled with respect to the first. If the two sequences are interchanged, the sign of  $\tau$  is changed. The autocorrelation function is computed in the same way as the correlation function. Since interchanging the two sequences in the autocorrelation function leaves the function unchanged, the autocorrelation function is an even function of  $\tau$ .

a	FIRST SEQUENCE	1 1 1 0 0 1 0 1 1 1 0
	SECOND SEQUENCE	0 1 0 1 1 1 0 0 0 1 0
	MOD 2 SUM	1 0 1 1 1 0 0 1 1 0 0

$$C = 5 - 6 = -1$$

$$C_n = (5 - 6)/(5 + 6) = 1/11$$

b	FIRST SEQUENCE	1 1 1 0 0 1 0 1 1 1 0
	SECOND SEQUENCE, $\tau=1$	0 0 1 0 1 1 1 0 0 0 1
	MOD 2 SUM	1 1 0 0 1 0 1 1 1 1 1

$$C = 3 - 8 = -5$$

$$C_n = (3 - 8)/(3 + 8) = -5/11$$

## CORRELATION FUNCTION

$\tau$	$C$	$C_n$	$\tau$	$C$	$C_n$
0	-1	-1/11	6	3	3/11
1	-5	-5/11	7	3	3/11
2	-1	-1/11	8	-1	-1/11
3	7	7/11	9	-1	-1/11
4	-5	-5/11	10	-1	-1/11
5	-1	-1/11			

Fig. 2. Correlation computation

These correlation concepts may be applied to codes. In contradistinction to sequences, *codes* are periodic functions of time that extend indefinitely far along the time axis. The correlation of two codes that have the same period  $T$  is defined as:

$$C = \int_0^T F_1(t) F_2(t) dt$$

If the two codes have different periods  $T_1$  and  $T_2$  the correlation is defined as

$$C = \int_0^{T_1 T_2} F_1(t) F_2(t) dt$$

Similarly, the normalized correlations are defined as

$$C_n = \frac{1}{T} \int_0^T F_1(t) F_2(t) dt$$

and

$$C_n = \frac{1}{T_1 T_2} \int_0^{T_1 T_2} F_1(t) F_2(t) dt$$

The correlation functions are defined as

$$C(\tau) = \int_0^T F_1(t) F_2(t + \tau) dt$$

and

$$C(\tau) = \int_0^{T_1 T_2} F_1(t) F_2(t + \tau) dt$$

The normalized correlation functions are defined as

$$C_n(\tau) = \frac{1}{T} \int_0^T F_1(t) F_2(t + \tau) dt$$

and

$$C_n(\tau) = \frac{1}{T_1 T_2} \int_0^{T_1 T_2} F_1(t) F_2(t + \tau) dt$$

Because of the periodicity of the codes, these correlation functions are equal to the correlation function usually defined for functions of time, namely:

$$R = \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T F_1(t) F_2(t + \tau) dt$$

The correlation function of two codes with the same period is periodic with the same period as the code. If two codes have periods which are relatively prime (i.e., the periods have no common divisors other than 1), the correlation is periodic with a period equal to the product of the periods of the codes. The period of the correlation function of two codes whose periods have a common divisor is the least common multiple of the periods of the codes.

The sequences used in constructing acquirable codes are PN sequences which have two level autocorrelation functions with the in-phase correlation  $p$  and the out-of-phase correlations  $-1$ . More formally:

$$C(\tau) = \begin{cases} p, & \tau = np, n = 0, \pm 1, \pm 2, \dots \\ -1, & \tau \neq np, n = 0, \pm 1, \pm 2, \dots \end{cases}$$

The correlation function for a code corresponding to a PN sequence is of the form shown in Fig. 3. These codes are called PN codes.

The correlation function of two PN sequences of relatively prime periods is everywhere small. Two such sequences are said to be uncorrelated. An example of the calculation of such a correlation function is given in Fig. 4. This calculation illustrates several important properties of this type of correlation function. First, the calculation need be performed only  $p_1$  times where  $p_1$  is the length of the shorter sequence; since, obviously, if the shorter sequence is shifted  $p_1$  times it returns to its original position. This is illustrated in Fig. 4 by having only 3 computations. Second, since the two sequences take on all possible phase relations with each other, the shifting of one code with respect to the other is equivalent to shifting the origin of both sequences by some amount. In Fig. 4 the relationship between the two sequences for  $\tau = 1$  is the same as that for  $\tau = 0$  except that the origin is shifted by 8. Similarly, the relationship between the two sequences for  $\tau = 2$  is the same as that for  $\tau = 0$  except that the origin is

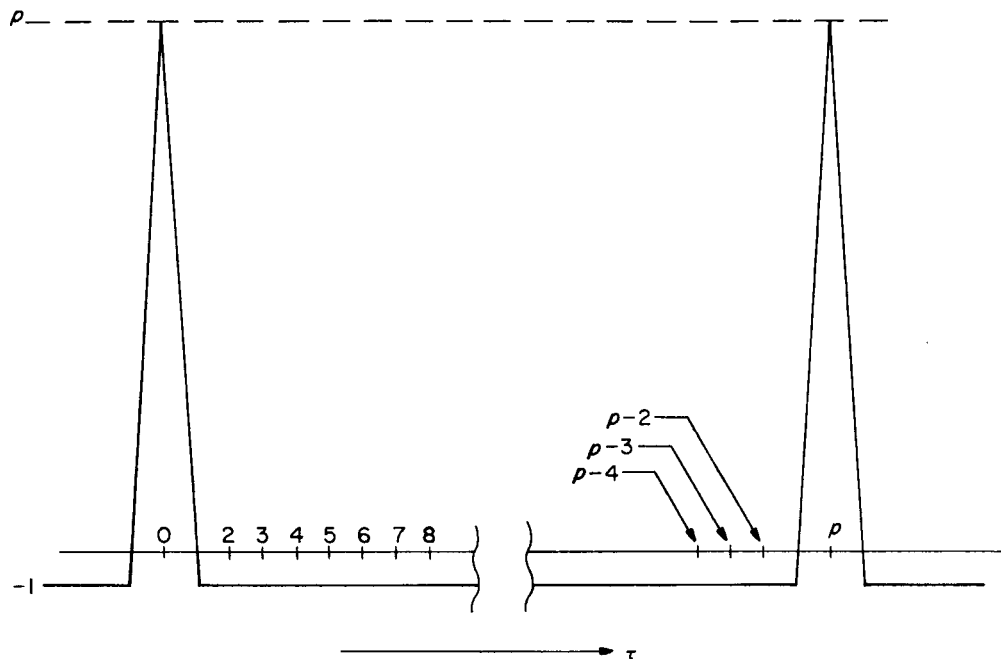


Fig. 3. Correlation function of PN code

shifted by 16. Since the correlation function is a sum (or integral) over a period, it is independent of the origin of the sequences. Therefore, as shown in Fig. 4, the correlation function of two PN sequences with relatively prime periods is a constant. The correlation function of two PN codes with relatively prime periods is also a constant.

3. *Statistical independence and probability.* Before developing the theory of the correlation functions of combined codes it is desirable to develop some more computational aids. To do this requires definitions of statistical independence and probability as applied to the digits in sequences. In a PN sequence of length  $p$  there are either  $p - 1/2$  zeros and  $p + 1/2$  ones or vice versa. In this section it will be assumed that there is one more *one* than *zero*. Therefore, if the sequence is sampled randomly, the probabilities of obtaining a *zero* or a *one* are:

$$P(0) = \frac{p-1}{2p}$$

$$P(1) = \frac{p+1}{2p}$$

Furthermore, if an entire sequence is examined, the relative frequencies of *zeros* and *ones* are precisely those given by the above probabilities. Since two sequences with relatively prime periods take on all phases with respect to each other during their combined period, knowing the phase of one sequence does not give any information

FIRST SEQUENCE	1 1 1 0 1 0
SECOND SEQUENCE	1 1 0

CALCULATION OF CORRELATION FOR $\tau = 0$	
FIRST SEQUENCE	1 1 1 0 1 0 0 1 1 1 0 1 0 0 1 1 1 0 1 0 0
SECOND SEQUENCE	1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0
MOD 2 SUM	0 0 1 1 0 0 1 0 1 0 1 1 1 1 1 0 0 0 0 1 0
NUMBER OF ZEROS = 11	
NUMBER OF ONES = 10	

CALCULATION OF CORRELATION FOR $\tau = 1$	
FIRST SEQUENCE	1 1 1 0 1 0 0 1 1 1 0 1 0 0 1 1 1 0 1 0 0
SECOND SEQUENCE	0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1
MOD 2 SUM	1 0 0 0 0 1 0 0 0 1 1 0 0 1 0 1 0 1 1 1 1
NUMBER OF ZEROS = 11	
NUMBER OF ONES = 10	

CALCULATION OF CORRELATION FOR $\tau = 2$	
FIRST SEQUENCE	1 1 1 0 1 0 0 1 1 1 0 1 0 0 1 1 1 0 1 0 0
SECOND SEQUENCE	1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1
MOD 2 SUM	0 1 0 1 1 1 1 1 0 0 0 0 1 0 0 0 1 1 0 0 1
NUMBER OF ZEROS = 11	
NUMBER OF ONES = 10	

Fig. 4. Calculation of the correlation function of two PN sequences

concerning the phase of the other sequence. More particularly, if two sequences are sampled randomly, the occurrence of a particular digit in one sequence does not affect the probabilities of the digits in the other sequence. Therefore, the two sequences are said to be statistically independent. One of the properties of statistical independence between two processes is that joint probabilities are the products of individual probabilities. This property may be used in the computation of correlation functions. Such a computation is shown in Fig. 5 for the same sequences as were used in Fig. 4. Note that the probabilities which are average frequency functions lead directly to the values for the normalized correlation functions. A convenient way of calculating the joint probabilities,

FIRST SEQUENCE      1   1   1   0   1   0   0  
 SECOND SEQUENCE    1   1   0

PROBABILITIES FOR FIRST SEQUENCE     $\begin{cases} P(0) = 3/7 \\ P(1) = 4/7 \end{cases}$

PROBABILITIES FOR SECOND SEQUENCE    $\begin{cases} P(0) = 1/3 \\ P(1) = 2/3 \end{cases}$

JOINT PROBABILITIES     $\begin{cases} P(00) = 3/7 \times 1/3 = 3/21 \\ P(01) = 3/7 \times 2/3 = 6/21 \\ P(11) = 4/7 \times 2/3 = 8/21 \\ P(10) = 4/7 \times 1/3 = 4/21 \end{cases}$

PROBABILITY OF AGREEMENT =  $P(00) + P(11) = 3/21 + 8/21 = 11/21$

PROBABILITY OF DISAGREEMENT =  $P(01) + P(10) = 6/21 + 4/21 = 10/21$

NORMALIZED CORRELATION =  $11/21 - 10/21 = 1/21$

	0	1
0	0	1
1	1	0

MOD 2 SUM

	3/7	4/7
1/3	3/21	4/21
2/3	6/21	8/21

PROBABILITIES

Fig. 5. Calculation of the correlation function of two PN sequences by probabilities

especially for more complicated situations, is to make use of a Karnaugh chart. The chart at the left in Fig. 5 is the Karnaugh chart for the mod 2 addition of two variables. The chart at the right corresponds to it, but the entries are the individual and joint probabilities. The probabilities of agreement and disagreement are obtained by summing the probabilities for zeros and ones, respectively.

4. *Correlation functions of combinations of sequences.* Some combinations of sequences have correlation functions which make them acquirable. A combination is formed by a logical function of two or more sequences digit by digit. An example of such a combination is shown in Fig. 6, where the logical *and* is the function used. There are four cases to be considered in computing the autocorrelation function of this sequence:

FIRST COMPONENT    1 1 1 0 1 0 0    SECOND COMPONENT    1 1 1 1 0 0 0 1 0 0 1 0 1 0

COMBINATION SEQUENCE    1 1 1 0 0 0 0 1 0 0 0 1 0 1 0 0 0 1 0 0 0 0 1 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 1 0

$p = p_1 p_2 = 7 \times 15 = 105$     1 0 1 0 0 1 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0

## CALCULATION OF PROBABILITY OF ONES AND ZEROS

IN THE COMBINATION SEQUENCE

Y	Y	
	0	1
X	0	0
	1	0

X	Y	
	7/15	8/15
3/7	21/105	24/105
	4/7	28/105

$$P(0) = (21 + 24 + 28)/105$$

$$= 73/105$$

$$P(1) = 32/105$$

## CALCULATION OF CORRELATION FOR THE SECOND COMPONENT

IN PHASE  $XY \oplus \bar{X}\bar{Y} = XY\bar{X} + \bar{X}YX$ 

Y	Xx			
	00	01	11	10
0	0	0	0	0
	1	0	1	1

Y	Xx			
	7/15	14/105	14/105	14/105
8/15	7/105	16/105	16/105	16/105
	2/7	2/7	2/7	2/7

$$P(0) = (7 + 14 + 14 + 8 + 16)/105 = 73/105$$

$$P(1) = (16 + 16)/105 = 32/105$$

$$C_n = (73 - 32)/105 = 41/105$$

## CALCULATION OF CORRELATION FOR THE FIRST COMPONENT

IN PHASE  $XY \oplus \bar{X}\bar{Y} = XY\bar{Y} + \bar{X}\bar{Y}Y$ 

X	Yy			
	0	01	11	10
0	0	0	0	0
	1	0	1	1

X	Yy			
	3/15	4/15	4/15	4/15
3/7	9/105	12/105	12/105	12/105
	4/7	12/105	16/105	16/105

$$P(0) = (9 + 12 + 12 + 12 + 16)/105 = 73/105$$

$$P(1) = (16 + 16)/105 = 32/105$$

$$C_n = (73 - 32)/105 = 41/105$$

## CALCULATION OF CORRELATION FOR NEITHER COMPONENT

IN PHASE  $XY \oplus \bar{X}\bar{Y} = XY\bar{X} + X\bar{Y}Y + \bar{X}YX + \bar{X}\bar{Y}Y$ 

$Xx$	$Yy$				$Xx$	$Yy$			
	00	01	11	10		3/15	4/15	4/15	4/15
00	0	0	0	0	1/7	3/105	4/105	4/105	4/105
01	0	1	1	0	2/7	6/105	9/105	8/105	8/105
11	0	1	0	1	2/7	6/105	8/105	8/105	8/105
10	0	0	1	1	2/7	6/105	8/105	8/105	8/105

$$P(0) = 57/105$$

$$P(1) = 48/105$$

$$C_n = (57 - 48)/105 = 9/105$$

Fig. 6. Calculation of the autocorrelation function of a combination sequence

- (1)  $\tau = 0$
- (2)  $\tau = n7$ , so that the first component is in phase
- (3)  $\tau = n15$ , so that the second component is in phase
- (4)  $\tau = \text{some other value}$ , so that neither component is in phase

The value of the correlation function for each of these cases can be computed using the Karnaugh chart and probabilities by first computing the mod 2 sum for the several cases. In the following equations capital letters are used for the components of the first combination and lower case letters are used for the components of the second combination. When two components are in phase, they are both designated by capital letters.

- (1)  $XY \oplus XY = 0$
- (2)  $XY \oplus Xy = XY\bar{y} + X\bar{Y}y$
- (3)  $XY \oplus xY = XY\bar{x} + \bar{X}Yx$
- (4)  $XY \oplus xy = XY\bar{x} + X\bar{Y}\bar{y} + \bar{X}xy + \bar{Y}xy$

If an attempt is made to compute the correlation functions for this combination using Karnaugh charts and probabilities as before, it is found that things do not work out, because components of the same length are not statistically independent. If the components are out of phase, then there are four cases: 00, 01, 11, and 10. If  $p$  is the length of the component, then the following statements can be made:

- (1) Number of 00 plus number of 11  $= \frac{p-1}{2}$
- (2) Number of 01 minus number of 10  $= \frac{p+1}{2}$
- (3) Number of 00 plus number of 01  $= \frac{p-1}{2}$
- (4) Number of 10 plus number of 11  $= \frac{p+1}{2}$
- (5) Number of 01 plus number of 11  $= \frac{p+1}{2}$
- (6) Number of 00 plus number of 10  $= \frac{p-1}{2}$

From these statements the following probabilities can be computed:

$$P(00) = \frac{p-3}{4p}$$

$$P(01) = P(10) = P(11) = \frac{p+1}{4p}$$

These probabilities apply for all PN sequences of length greater than 3. The sequence of length 3 is short enough to be degenerate, which is why it is not used in the example in Fig. 6.

There are two points that should be noted about the correlation function computed in Fig. 6. First, there are peaks for all values of  $\tau$  which are divisible by the period of either component. Second, the correlation when both components are out of phase is not  $-1/p_1p_2$ . This is a consequence of the fact that the combination sequence is very unbalanced with 73 zeros to 32 ones.

The computation in Fig. 6 can be simplified by making an approximation; namely, that a PN sequence has equally many *ones* and *zeros*. This leads to the approximation that for an out-of-phase component, the four situations, 00, 01, 11, and 10, occur equally often. The computation for the same sequences as were used in Fig. 6 is repeated in simplified form in Fig. 7. The case where neither component is in phase is done first. Note that *with* the approximation, all of the probabilities in the Karnaugh chart are equal; so that one might as well merely count the number of *ones* and *zeros* in the chart and use the formula:

$$C_n = \frac{\text{number of zeros} - \text{number of ones}}{\text{number of zeros} + \text{number of ones}}$$

The error in this computation is:

$$\text{error} = \frac{1}{4} - \frac{9}{105} = \frac{105 - 36}{420} = 0.164$$

This is a fairly large error, but it decreases rapidly with the length of the component sequences. Aside from the case of computation, the approximate method has another advantage; it applies to codes of any length.

The computation of the autocorrelation for one component in phase is given in Fig. 7 also. It is not necessary to perform the other computation, since the two components enter into the combining expression

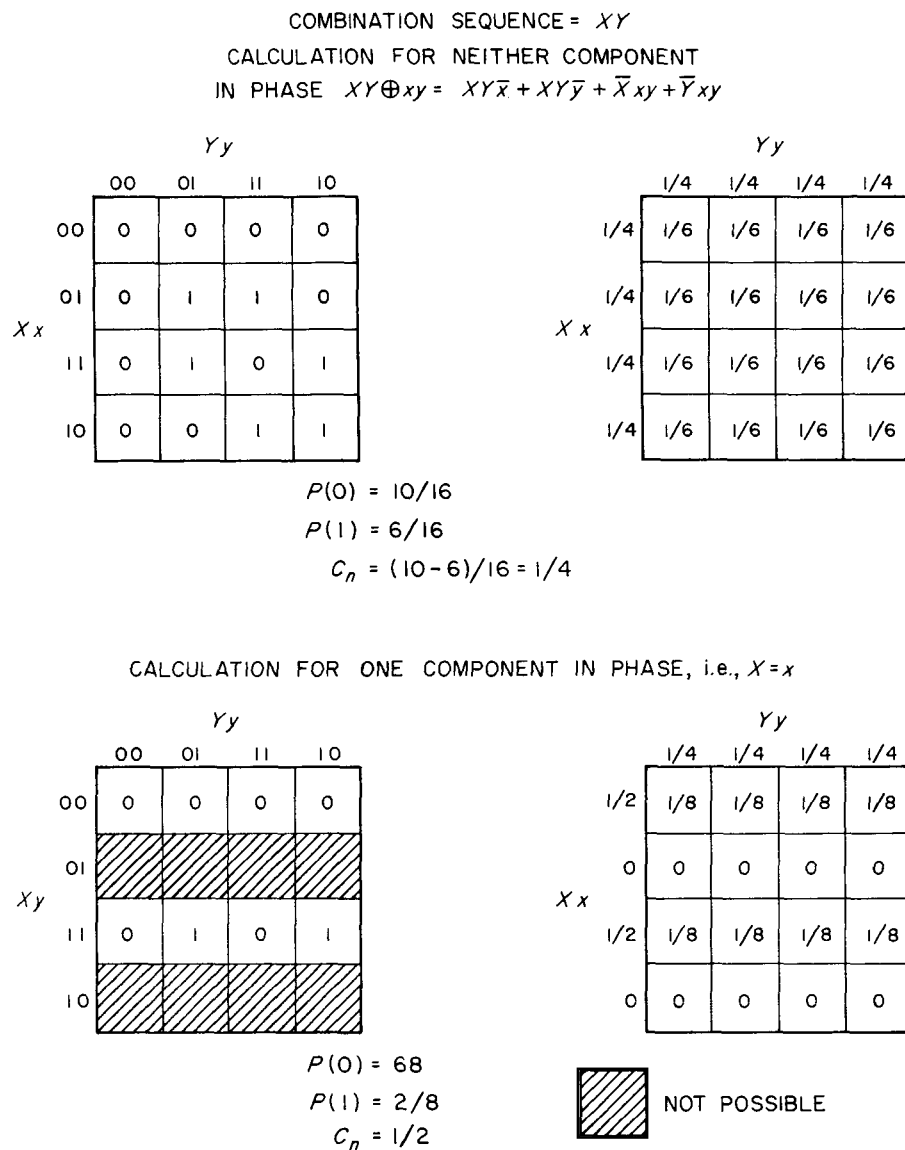


Fig. 7. Simplified calculation of the autocorrelation function of a combination sequence

symmetrically. Also, the complete Karnaugh chart can be used by merely deleting those parts which are ruled out by two of the variables being the same. This is equivalent to making the probabilities for those cases zero as shown in the chart on the right in Fig. 7. The error in this computation is:

$$\text{error} = \frac{1}{2} - \frac{41}{105} = 0.228$$

Again the error is large, but it decreases with increasing length of the sequences. Because of the simplicity and generality of the approximate computation, it will be used in the rest of this paper. Of course, in an actual application the precise correlations should be calculated.

Not all combination sequences have autocorrelation functions with peaks at multiples of the periods of the component sequences. An example of such a combination is given in Fig. 8. In this combination both components must be in phase for the correlation to be other than very small. This combination then is equivalent to a single PN sequence as far as its correlation properties are concerned. The sequence shown in Fig. 7 is said to be acquirable, while that in Fig. 8 is not.

COMBINATION SEQUENCE =  $X \oplus Y$

CALCULATION FOR NEITHER COMPONENT  
IN PHASE  $X \oplus Y \oplus X \oplus Y$

$Y \ Y$

		00	01	11	10	
$X \ X$	00	0	1	0	0	
	01	1	0	1	0	
	11	0	1	0	1	
	10	1	0	1	0	

$C_n = (8 - 8) / 16 = 0$

IF  $X=x$ , THE SECOND AND FOURTH ROWS ARE  
DELETED AND  $C_n = (4 - 4) / 8 = 0$

IF  $Y=y$ , THE SECOND AND FOURTH COLUMNS  
ARE DELETED AND  $C_n = (4 - 4) / 8 = 0$

Fig. 8. Autocorrelation function of a combination which has no extra peaks

Another correlation function of interest is that of a combination sequence vs one or more of its components. An example of this type of correlation is given in Fig. 9.

COMBINATION SEQUENCE =  $XY$ 

$$XY \oplus x = XY\bar{x} + \bar{X}x + Yx$$

		$XX$			
		00	01	11	10
$Y$	0	0	1	1	0
	1	0	1	0	1

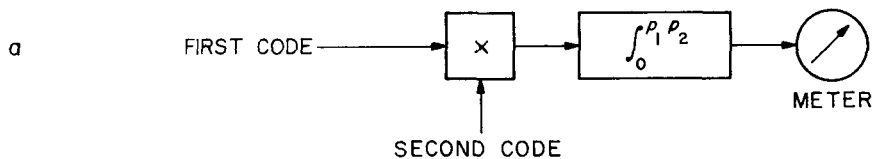
$$C_n = 0$$

IF  $X = x$ , THE SECOND AND FOURTH ROWS ARE DELETED AND  $C_n = (3-1)/4 = 1/2$

Fig. 9. Calculation of the correlation function of a combination with one of its components

## B. Correlation and Tracking Schemes

1. *Correlators and detectors.* When the problem of mechanizing a device to determine the correlation of two codes is considered, perhaps the most obvious solution is that shown in Fig. 10a. Such a device works perfectly well if the two inputs are simply the two codes. In fact, because of the binary nature of the codes, the multiplier



SIMPLE CORRELATOR

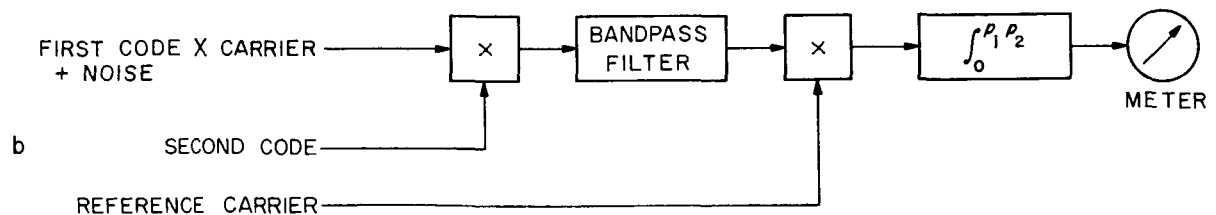


Fig. 10. Correlation detector for a noisy signal

can be a switching device or even a digital device. However, in practical cases in a ranging system, the one code is badly contaminated with noise. The total noise power at the input to the multiplier may be as much as 50 db above the signal power. It is extremely difficult to build wideband multipliers with dc outputs to work under these conditions. If the code is available as modulation on a carrier, the multiplier is much easier to build. Figure 10b shows a possible form for a correlator where the code is available as biphasic modulation on a carrier. The bandpass filter eliminates much of the noise to the one input of the second multiplier, both of whose inputs are narrow band and may be tuned. The phases of the carriers are arranged so that when two digits of the codes are the same, the carrier out of the first multiplier (actually at the output of the bandpass filter) is in phase with the reference to the second multiplier. When two corresponding digits are not the same, the carrier out of the bandpass filter is reversed. Thus, if the two codes are correlated, the input to the second multiplier is always in phase with the reference. If the codes are uncorrelated, the signal to the second correlator is in phase with the reference half the time and out of phase half the time, so that the integrated output of the multiplier is zero. For partial correlation, the output of the integrator is proportional to the correlation.

2. *Direct tracking.* Another problem of interest in ranging is tracking a code once it has been acquired. For purposes of discussion, assume that a single component, i.e., a PN sequence, has been transmitted to an object and returned, and that, by measuring the correlation between the returned signal and a local model, the local model has been brought into phase with the returned signal. Now the phase difference between the transmitted signal and the local signal is a measure of the range. However, if the range changes, the local model is no longer in step with the returned signal. In order to make a continuous range measurement it is desirable to have a *tracking* device, which would automatically keep the local model in step with the returned signal.

A correlation detector such as that shown in Fig. 10b is suitable for acquiring the returned signal (i.e., matching a local model to it), but it does not lend itself to automatic tracking because, as it stands, it does not generate an error signal when the two codes tend to go out of phase. A scheme for generating an error signal is shown in Fig. 11. Once code *B* is in phase with code *A*, any tendency for it to move out of phase will generate an error signal which is negative if code *B* gets ahead and positive if code *B* gets behind. This error signal can be used to control the rate at which code *B* is switched and thereby keep code *B* in phase with code *A*. This scheme has the same difficulty that the correlator in Fig. 10a has in the presence of noise; viz., the multipliers are difficult to build.

3. *Tracking with a carrier or clock.* The correlation detector in Fig. 10b can be made the basis of a tracking device by attaching a phase-locked loop to track the carrier and using the carrier to clock the generator of the local code. The form of such a system is shown in Fig. 12. The inner loop is a phase-locked loop which tracks the clock.

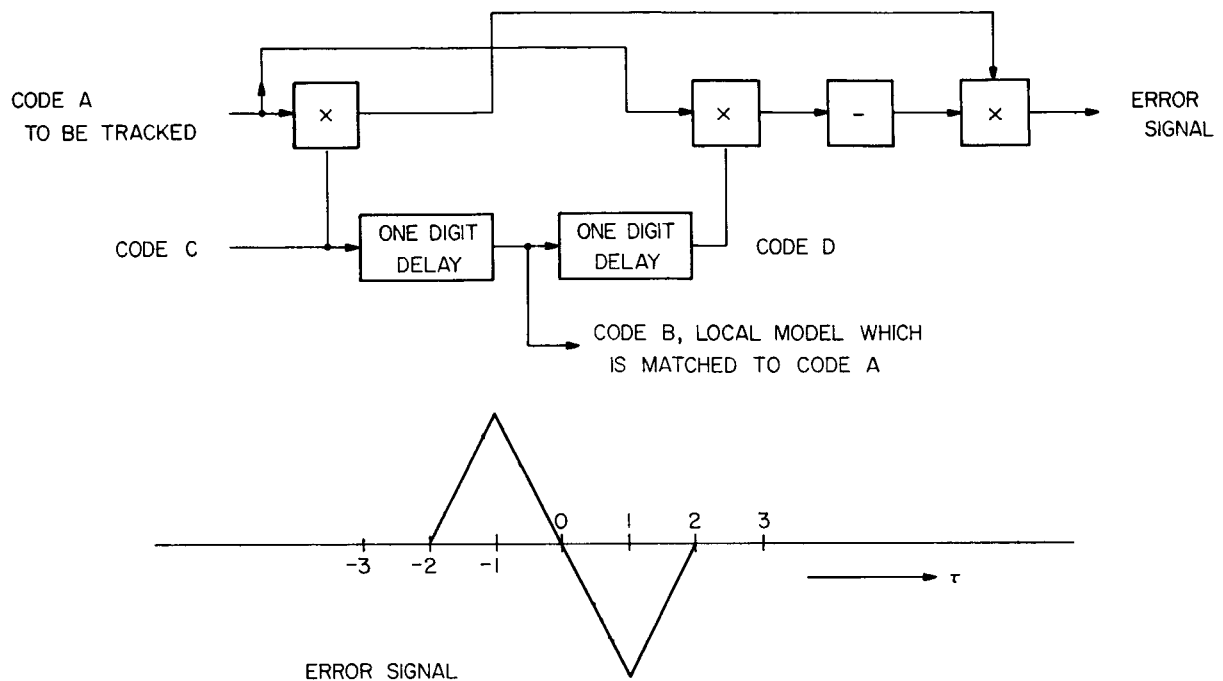


Fig. 11. Method of generating an error signal for tracking a code

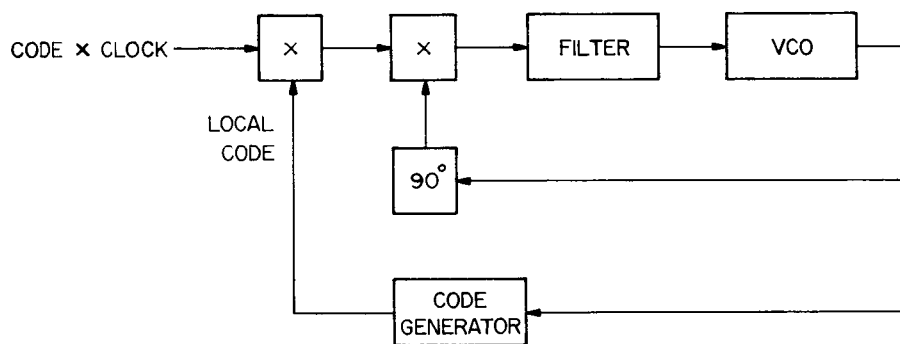


Fig. 12. Double-loop code tracking device

The outer loop forces the local code to follow the clock. The clock in the incoming signal is the clock that was used to generate the incoming code so that they stay in phase. Suppose that the two codes are exactly in phase and that the output of the VCO is exactly in phase with the incoming clock. Then there is no dc signal at the output of the second multiplier. If the incoming code and clock tend to shift with respect to the local code and clock, there is a dc signal at the output of the second multiplier which causes the VCO to shift frequency and brings the local clock and hence the local code back into phase with the incoming signals.

This two-loop tracking device is the heart of the ranging system receiver. In the ranging system the incoming code is acquired to establish the round trip time and is tracked to keep the measure of the round trip time continuously up to date.

4. *Clock frequencies and error curves.* In the previous section the frequency of the clock relative to the switching frequency of the code was not mentioned. This is an important parameter. Before discussing the frequency of the clock, note that the error signal is a function of  $\tau$ . Specifically,

$$E(\tau) = \int_{\text{period}} \text{code}_1(t) \text{clock}(t) \text{code}_2(t + \tau) \text{clock} \left( t + \frac{P}{4} + \tau \right) dt$$

In this case the integral can be approximately factored into two parts:

$$E(\tau) \approx \left[ \int_{\text{period}} \text{code}_1(t) \text{code}_2(t + \tau) dt \right] \times \left[ \int_{\text{period}} \text{clock}(t) \text{clock} \left( t + \frac{P}{4} + \tau \right) dt \right]$$

But these two integrals are just the correlation functions of the codes and of the clocks. Figure 13a shows the correlation function of a clock which is chosen to be a square wave for convenience in sketching the function, but which might equally well be a sine wave. Figure 13b shows the correlation function for a PN code which has a period of the clock wave. Unfortunately, the resulting error function has six stable null points. This makes it unsuitable for the double-loop tracking device, since in actual operation it is not possible to predict which of the nulls will be the operating point.

A more suitable clock is one which has a period equal to two digit periods of the code. The error curve of such a code and clock is shown in Fig. 14, which also shows another way of looking at the error curve. The double product can be written as

$$(\text{code}_1 \times \text{clock}_1) (\text{code}_2 \times \text{clock}_2)$$

The error curve is then exactly the correlation function of the two waves shown in Fig. 14. This function makes a good error curve in that there is only one stable null near  $\tau = 0$ . There are two quasi-stable nulls near the odd half periods of the correlation function. Sometimes these are troublesome in a system and sometimes they are not.<sup>1</sup> One way of coping with this problem is described in the next section.

<sup>1</sup> A way of inverting the correlation function in the vicinity of the odd half periods has recently been suggested by J. Springett (Jet Propulsion Laboratory).

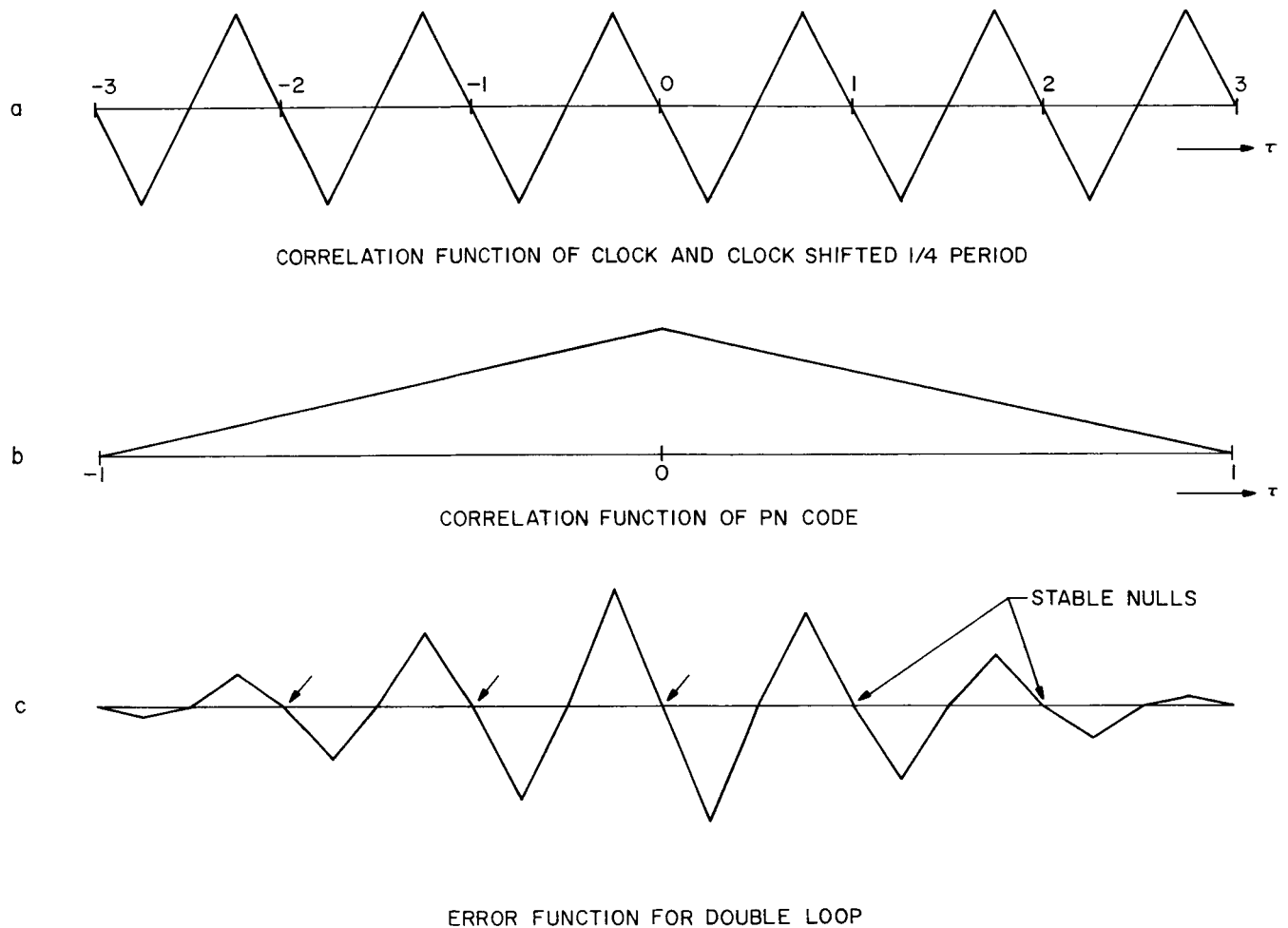


Fig. 13. Computation of error function for double-loop high-frequency clock

The existence of the inverted portion of the correlation function at  $\tau = p_1 p_2 / 2$  is due to the fact that at this point the PN components are in phase while the clock components are half a period out of phase relative to the phases at  $\tau = 0$ .

### C. Acquirable Codes

1. *Single component with clock.* Individual methods for acquiring several types of codes will be given in this section. The code situations are listed in increasing order of complexity. Perhaps the simplest acquirable code is the PN code itself. Certainly the PN code times clock described in the previous section is acquirable if some provision is made to avoid locking the loop at one of the quasi-stable nulls. However, such a code is not useful for ranging.

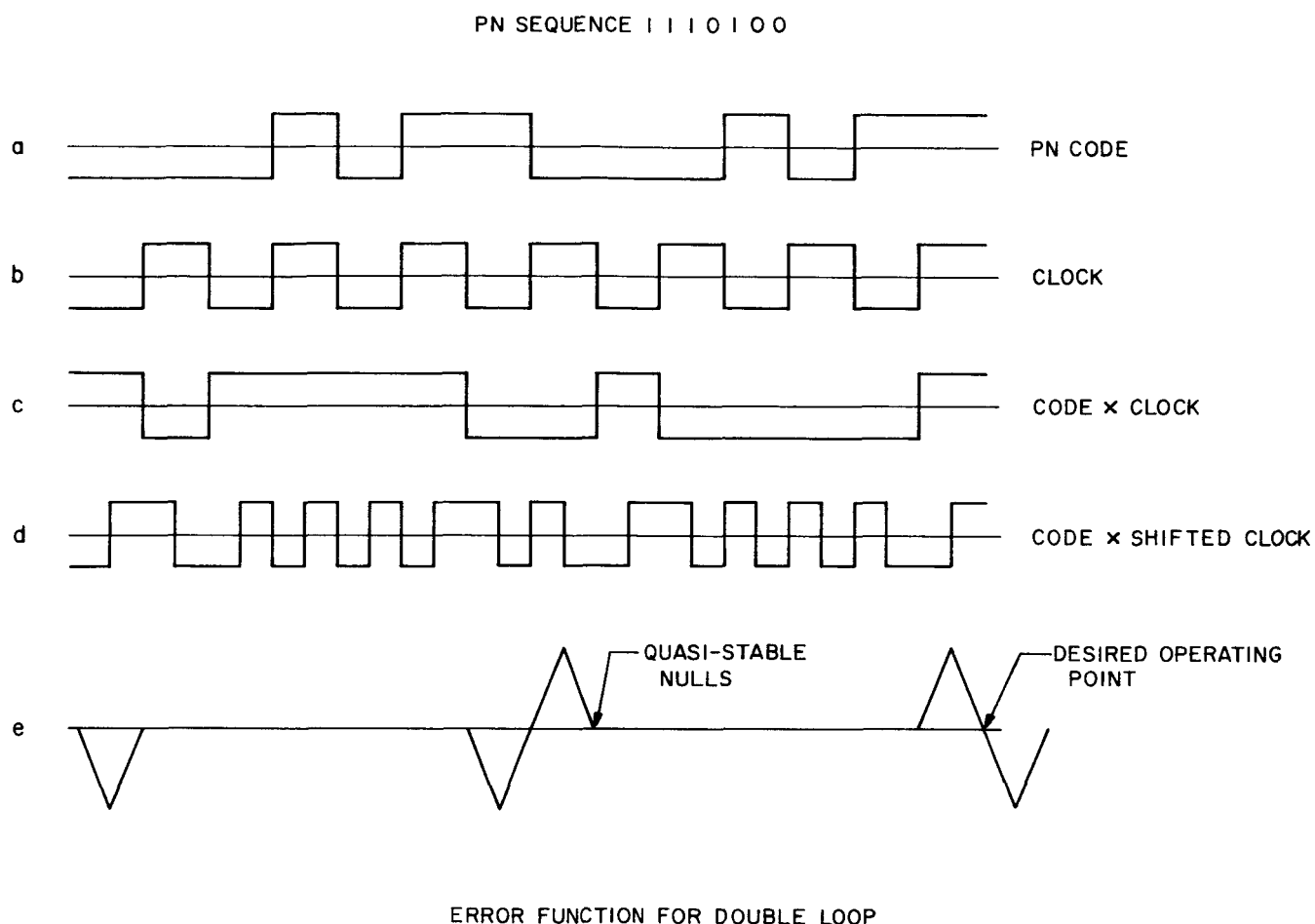


Fig. 14. Computation of error function for double-loop best clock frequency

2. *Combined acquirable codes.* The first sequence that was selected for ranging used the combining function:

$$W = X \oplus YZ$$

The autocorrelation function of such a sequence is computed in Fig. 15. Notice that since the  $Y$  and  $Z$  components enter into the combining expression symmetrically, it is only necessary to compute for one of them. This autocorrelation function suggests a procedure for acquiring the code using a double-loop tracking device with some additional equipment as shown in Fig. 16. If, initially, the incoming clock has a frequency different from the output of the VCO, then the two codes will slip past each other continuously. If the rate of slippage is not too high, the phase-locked loop will lock up when the  $X$  components come into phase. (It is assumed that some means not specified is used to prevent locking on the quasi-stable nulls.) At this time the output of the VCO will be in phase with the incoming clock. Further, 62½% of the time the signal out of the first multiplier will be a clock in phase with the output of the

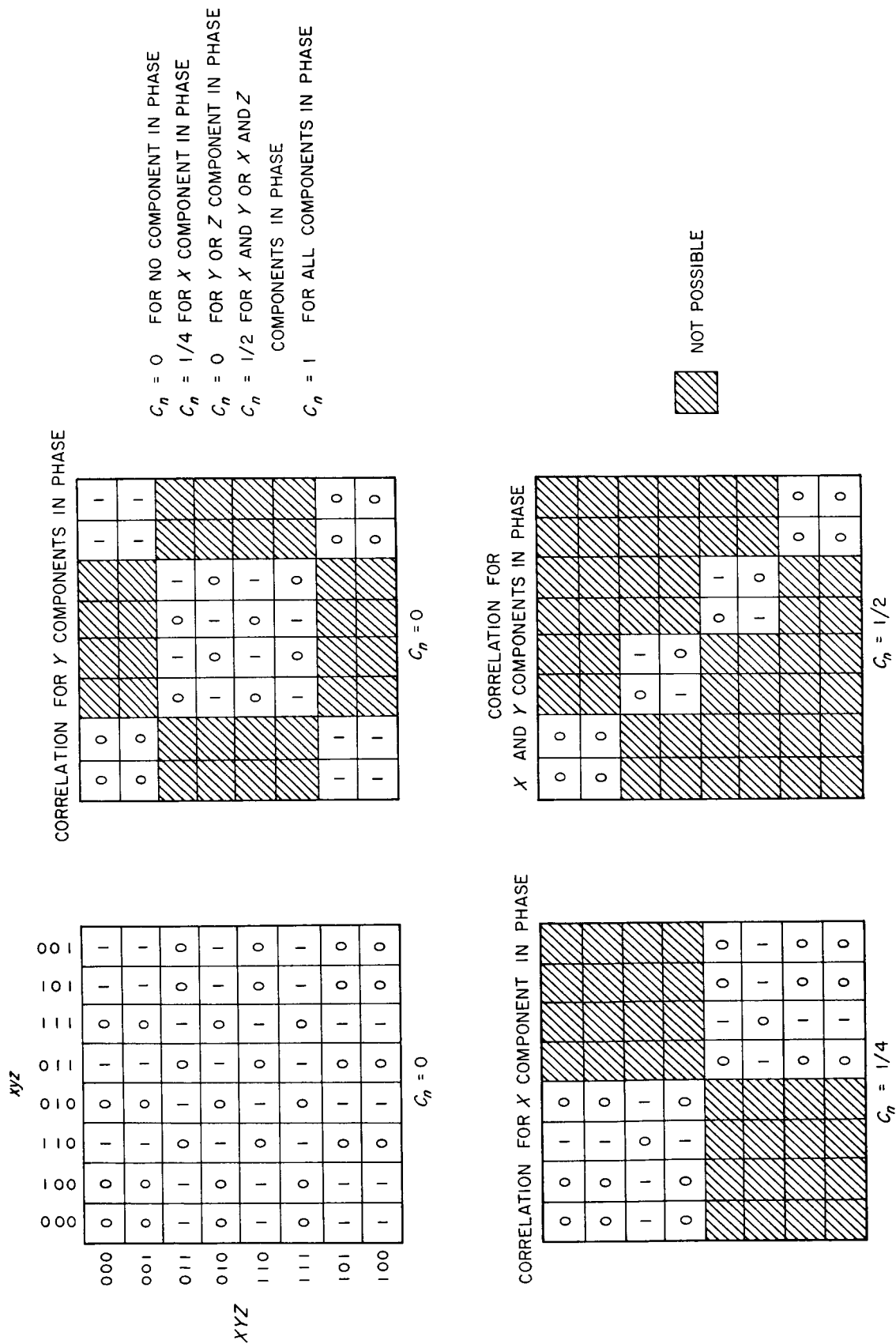


Fig. 15. Calculation of autocorrelation function for ranging sequences  
 $(X \oplus YZ) \oplus (x \oplus yz)$

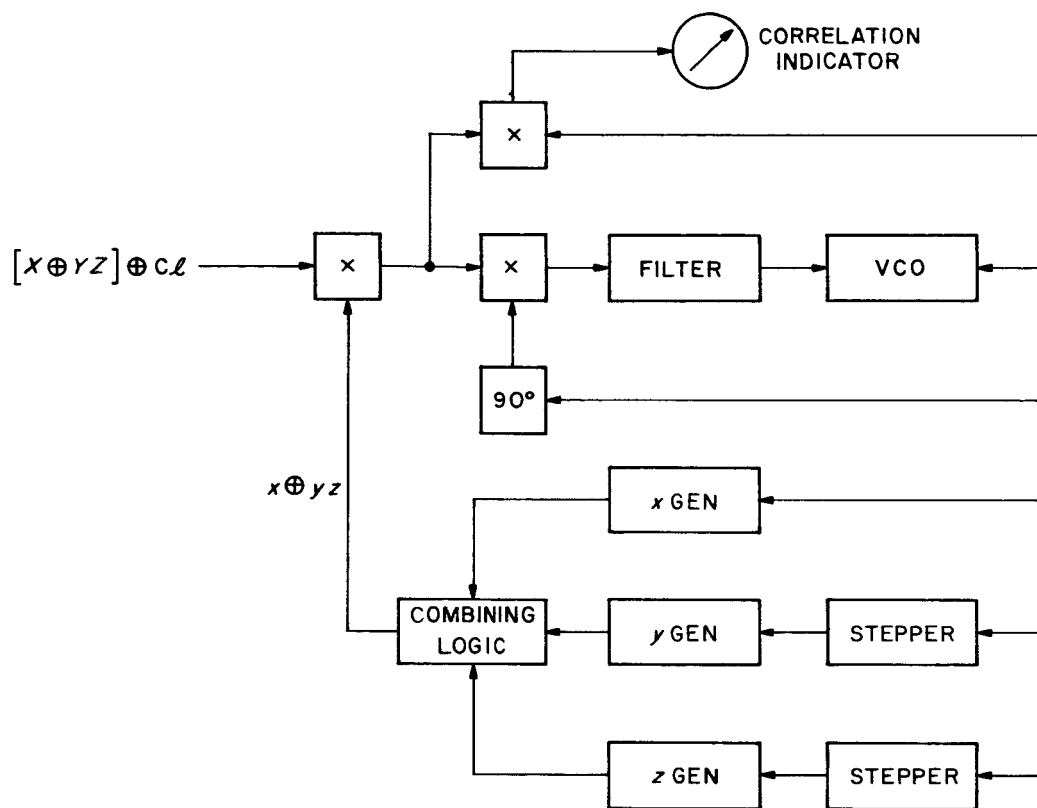


Fig. 16. System for acquiring a three-component code

VCO, while  $37\frac{1}{2}\%$  of the time it will be 180 deg out of phase. On the average, the output of the first multiplier may be considered as a clock in phase with the output of the VCO, but only 25% of full amplitude. This signal is synchronously detected and shown on the meter.

The second step in the procedure is to step the  $y$  generator a digit at a time until the two  $Y$  components are in phase. The meter, which is a correlation indicator, will then indicate a correlation of  $1/2$ . The third step in the procedure is to step the  $z$  generator until the two  $Z$  components are in phase. The meter will then indicate a correlation of 1.

The system just described has two disadvantages. First, and least important, is the problem of the spurious nulls, which can be overcome by additional complexity. The second disadvantage is that this system requires the phase-locked loop to lock during the interval when the two  $X$  components are in phase. This implies either a wide-band loop, which is undesirable when the noise level on the incoming signal is high, or a very slow sweeping rate, i.e., only a very slight initial difference in frequencies. In the ranging situation either of these alternatives is very unfavorable. In other applications this may not be the case.

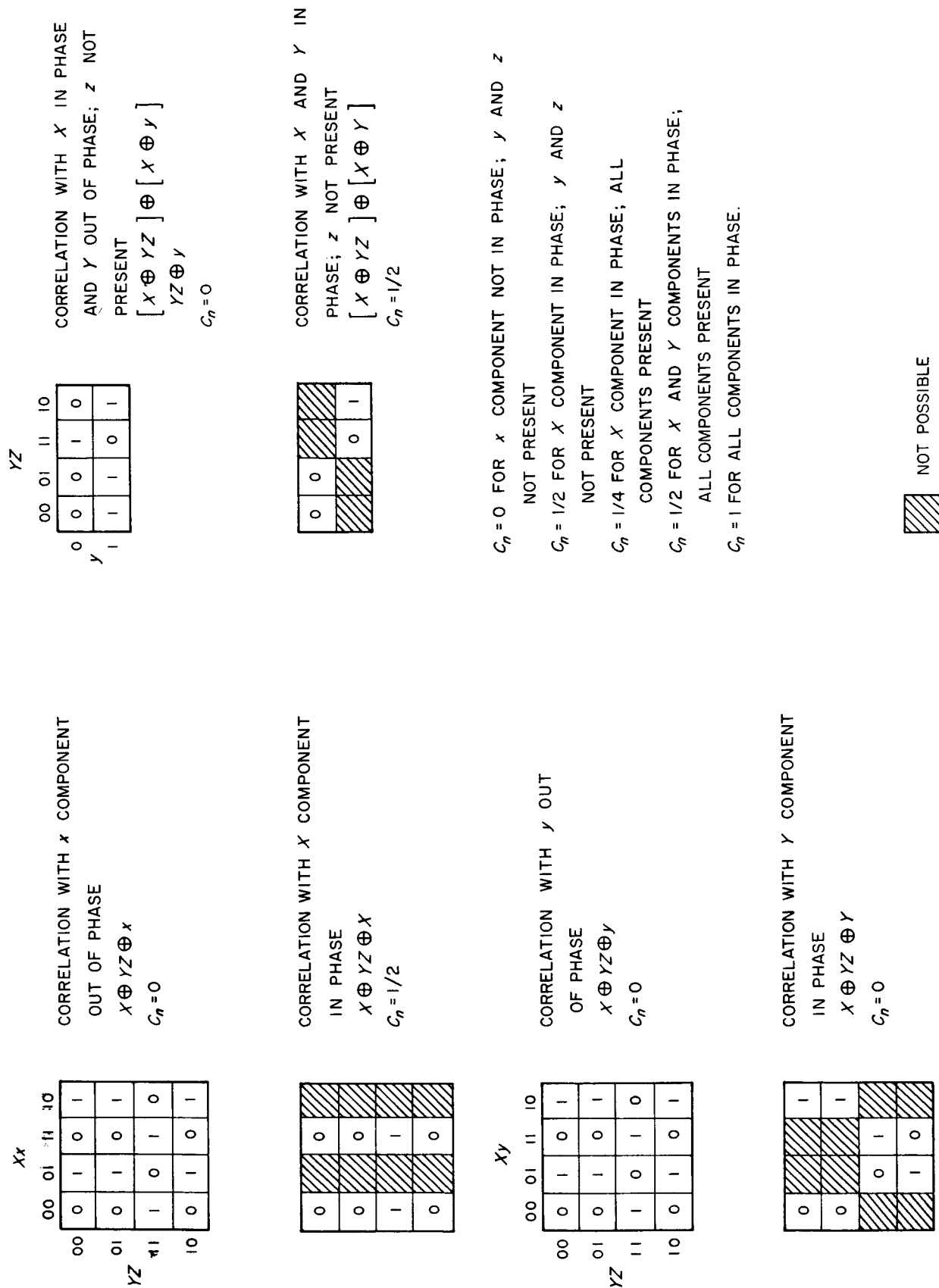


Fig. 17. Correlation of a combination sequence with its components  
(sequence =  $X \oplus YZ$ )

The system shown in Fig. 16 can be used to illustrate another scheme for acquisition which is somewhat better than the one described. This makes use of the correlation function of a code with its components. The correlation of the sequence  $W = X \oplus YZ$  with its components is computed in Fig. 17 and the interesting correlations listed. This leads to a procedure in which only the  $X$  component is correlated in the first step and the correlation when the  $X$  component and the clock are acquired is  $1/2$  instead of  $1/4$ . This higher signal has an advantage in that it makes it easier for the phase-locked loop to lock up. However, the loop must still lock during the interval during which the two  $X$  components are in phase.

3. *Acquirable codes with a clock component.* The difficulty of locking up the clock loop on the fly, so to speak, has led to the development of combined codes in which one of the components is the clock. The clock can be considered as a sort of degenerate PN sequence of length 2. Its in-phase correlation is 2 and its out-of-phase correlation is  $-2$ . Unlike all other PN sequences it is perfectly balanced. An example of such a combined sequence is  $W = \text{clock} \oplus YZ$ . This can be interpreted as  $W = \text{clock}$  when  $YZ = 0$ , and  $W = \overline{\text{clock}}$  when  $YZ = 1$ .

Since  $YZ = 0$  occurs 75% of the time while  $YZ = 1$  occurs 25% of the time, on the average  $W$  is a clock whose amplitude is 50% of the full amplitude. Another way of looking at the situation is to compute the correlation as seen by the indicator in Fig. 16 but to include the clock in the computation. Then in Fig. 16 the incoming signal is  $\text{clock} \oplus YZ$  and the local signal is merely  $YZ$ . The loop is free to lock up immediately and can slip any number of cycles in locking. The  $Y$  and  $Z$  components can then be acquired as before. There is no  $X$  component as such. The correlation indicator readings for each step in the procedure are:

(1) clock not locked	0
(2) clock locked	25%
(3) $Y$ locked	50%
(4) $Z$ locked	100%

This procedure has two very great advantages over previous procedures. First there is a clock signal for the phase-locked loop to lock on even when no component is matched. Second, there are no quasi-stable false nulls to which the system can lock. The correlation function of the clock and the clock shifted 90 deg is the triangular function in Fig. 13a which has no quasi-stable nulls.

4. *Methods of obtaining a clock component.* There are several methods which might be used to give a clock component. The first is to use the clock as one component of the code. A second would be to add, in an analog way, a clock signal to the transmitted signal. This is not very attractive in a situation where the codes are being generated

by digital equipment, but it does suggest that the same average result could be obtained by switching between two codes, one of which was a clock. For example, if it were decided to have a code which had a 50% clock component independently of the phases of the other component, one might have two digits of clock followed by two digits of code, etc. An example of this kind of a sequence is shown in Fig. 18. The corresponding sequence to produce the

$X$ COMPONENT	1 1 1 0 1 0 0 1 1 1 0 1 0 0
CLOCK	1 0 1 0 1 0 1 0 1 0 1 0 1 0
$X \oplus Cl$	0 1 0 0 0 0 1 1 0 1 1 1 1 0
$X \oplus Cl$	0 1   0 0   0 1   1 0   0 0   1 1   1 1
$Cl$	1 0   1 0   1 0   1 0   1 0   1 0   1 0
$X$	1 1   1 0   1 1   0 0   1 0   0 1   0 1
0	0 0   0 0   0 0   0 0   0 0   0 0   0 0
MOD 2 SUM	1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0

FULL AGREEMENT WITH CLOCK WHEN  $X$   
COMPONENT IS IN PHASE

$X \oplus Cl$	0 1   0 0   0 1   1 0   0 0   1 1   1 1
$Cl$	1 0   1 0   1 0   1 0   1 0   1 0   1 0
$x$	0 1   0 1   1 1   1 0   1 1   0 0   1 0
0	0 0   0 0   0 0   0 0   0 0   0 0   0 0
MOD 2 SUM	0 0   0 1   1 0   0 0   1 1   1 1   0 1
	1 0   1 0   1 0   1 0   1 0   1 0   1 0

AGREEMENTS = 6

DISAGREEMENTS = 8

$$C_n = -1/7$$

Fig. 18. Example of alternating PN and clock

local code would have zeros during the clock digits and the  $X$  component during the other digits. The bit-by-bit correlation of these two alternating sequences is shown for the cases when the  $X$  component is in phase and for a case when the  $X$  component is out of phase. This arrangement leads to 100% clock when the two  $X$  components are in phase and (approximately) 50% clock when the  $X$  component is out of phase. This is, of course, just what would be expected. It is not even necessary to consider the digits which are strictly clock in making the computation.

Although it is not obvious, perhaps, the only effect of sampling the  $X \oplus$  clock is to rearrange the digits. Since the two  $X$  components are both sampled in the same way, their digits are both rearranged in the same way and this does not affect the correlation properties.

#### D. Generation and Manipulation of Sequences

1. *Maximal length linear shift register sequences (see Ref. 2).* Any periodic sequence is generated by a recursion formula; i.e., the  $n$ th digit is produced by some function of the preceding  $i$  digits:

$$A_n = F(A_{n-1}, A_{n-2}, \dots, A_{n-i})$$

In the case of the maximal length shift register sequence the recursion formula is particularly simple,  $F$  being the modulo 2 sum of some of the preceding  $i$  digits, with  $i = \log_2(p+1)$ . The usual way of mechanizing the generation of such a sequence is shown in Fig. 19, where a sequence of length 31 is used for example. The shift register is

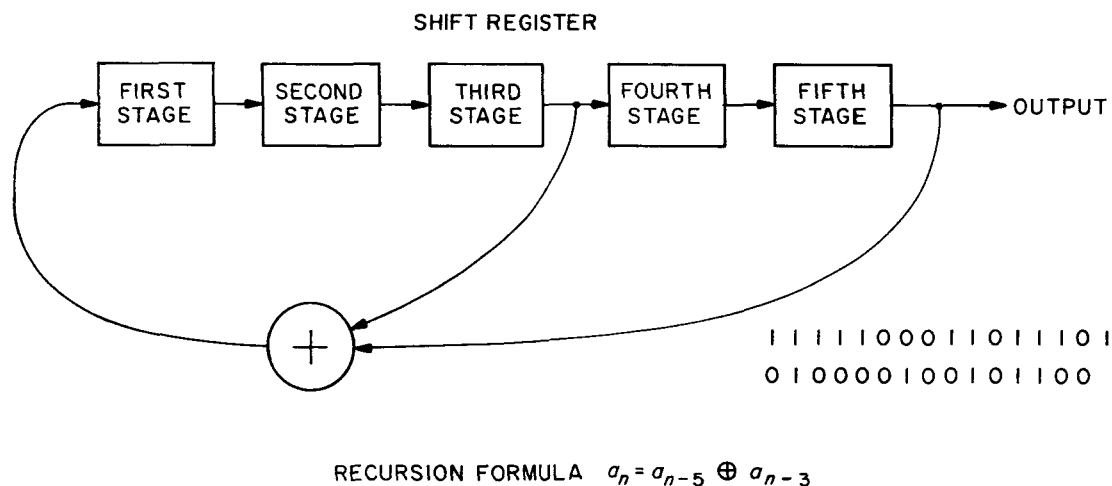


Fig. 19. Generator for maximal-length linear shift register sequence

used to store 5 consecutive digits. The mod 2 adder forms the sum of the digits in the third and fifth stages and feeds it to the input to the first stage. After the next shift of the shift register this digit is stored in the first stage. The sequence starting with *ones* in the shift register is shown to the right of the shift register and is read from left to right. At first glance it might seem that the sequence should appear at the output of the mod 2 adder rather than at the output of the last stage, and well it might. Actually, the sequence appears at the output of the adder and at each

stage, the only difference being a phase shift. The shift register with feedback is a kind of oscillator and has the same "waveform" at all points except for a phase shift. This is illustrated in Fig. 20, where the sequence as it appears at all six points is listed with the proper relative phases. Figure 20 illustrates another point. The digits

SEQUENCE AT FIRST STAGE	SEQUENCE AT SECOND STAGE	SEQUENCE AT THIRD STAGE	SEQUENCE AT FOURTH STAGE	SEQUENCE AT FIFTH STAGE	SEQUENCE AT FEEDBACK	NUMBER IN REGISTER
—	—	—	—	—	0	31
0	—	—	—	—	0	15
0	0	—	—	—	0	7
0	0	0	—	—	—	3
—	0	0	0	—	—	17
—	—	0	0	0	0	24
0	—	—	0	0	—	12
—	0	—	—	0	—	22
—	—	0	—	—	—	27
—	—	—	0	—	0	29
0	—	—	—	0	—	14
—	0	—	—	—	0	23
0	—	0	—	—	—	11
—	0	—	0	—	0	21
0	—	0	—	0	0	10
0	0	—	0	—	0	5
0	0	0	—	0	0	2
0	0	0	0	—	—	1
—	0	0	0	0	0	16
0	—	0	0	0	0	8
0	0	—	0	0	—	4
—	0	0	—	0	0	18
0	—	0	0	—	—	9
—	0	—	0	0	—	20
—	—	0	—	0	0	26
0	—	—	0	—	0	13
0	0	—	—	0	—	6
—	0	0	—	—	—	19
—	—	0	0	—	—	25
—	—	—	0	0	—	28
—	—	—	—	0	—	30
—	—	—	—	—	0	31
0	—	—	—	—	0	15

← INJECT EXTRA ONE TO SHIFT LEFT  
 ← INJECT EXTRA ONE TO SHIFT RIGHT

Fig. 20. Phases of maximal-length linear shift register sequence

stored in the shift register at any time may be considered to be a binary number. The decimal equivalent of the contents of the register is shown adjacent to each phase. All possible 5-digit binary numbers appear in the register

except zero. Since any linear (i.e., mod 2) combination of zeros yields a zero, that number is excluded. Because all the others are included, the sequence is as long as it is possible to have with a 5-stage shift register and linear feedback logic. Hence, the sequence is termed a maximal-length linear shift register sequence.

Maximal-length linear shift register sequences are PN sequences and therefore useable as components of acquirable sequences. The use of a component in an acquirable sequence requires that it be capable of being shifted both left and right. The convention chosen is that the time axis runs from left to right, past to future, so that shifting a sequence *right* one digit means that a given digit occurs one digit interval later than it otherwise would. If the shift register is not running near the maximum speed for the particular equipment being used, the sequence can be shifted by deleting or adding a shift. However, in some cases—for example, when the shift register is a delay line—this is not possible. It then becomes necessary actually to modify the sequence. To shift left it is necessary to delete a digit from the sequence. One way in which this may be done is to add a word detector which looks for a particular word in the shift register, and then injects an extra *one* into the mod 2 adder. If the word is properly chosen this will shorten the sequence by one digit. This is a particular case of a more general situation which has previously been described (Ref. 19). In the sequence shown in Fig. 19 and 20 the word detector should look for the word 11110. The next digit is normally a *one*, making the next word 11111. If an extra *one* is injected into the mod 2 adder, the next digit is a *zero*, and the next word is 01111. The total effect is to skip the word 11111. The *all ones* word is the appropriate word to be skipped in any sequence. Of course, if only one shift is desired, provisions must be made to effect this operation only once.

To shift right there are two possibilities: either repeat a word or add a word. The only word that can be repeated is the *all ones* word. This can be done by enabling a word detector that looks for the *all ones* and injects an extra *one* into the mod 2 adder. Again, this must be disabled after one operation if only one shift is desired. The only word that can be *added* is the *all zeros*. This requires two word detectors, one to look for the word 0...01 and cause the following word to be 0...0 instead of 10...0, and a second to look for the word 0...0 and cause the next word to be 10...0. The choice between the two methods is a matter of convenience in the mechanization. Usually the word detector to look for the *all zeros* word is necessary anyway to act as an automatic starter. If, when the shift register is turned on, all of the stages happen to have zeros stored, no sequence, or, equivalently, a sequence exclusively of zeros will be produced. A word detector to detect the *all zeros* and inject a *one* into the mod 2 adder will then act to start the sequence. This word detector is then available for use in the shifting process.

One other manipulation that is useful in a ranging system is synchronization. It is desirable to synchronize two sequence generators that are generating the same sequence. This can be done by feeding the output of one

feedback logic into both first stages. As soon as  $i$  digits have passed, the contents of the two registers will agree, and if the generator being synchronized is returned to normal operation it will continue in synchronism. Of course, the two generators must be clocked from the same source in order to be synchronized.

2. *Direct generation of other PN sequences.* PN sequences other than the maximal-length linear shift register sequences can also be generated by a shift register with feedback, but there are two differences. First, the sequences are not maximal length, so that proportionately more stages are required in the shift register; and second, since the sequences are not linear, the feedback logic may be more complicated. Therefore, in general, a generator for one of these sequences may be much more complicated than for a maximal-length linear shift register sequence of comparable length. Although a shift register with more stages is no particular problem, the feedback logic may be so complicated that a large computer operation is required to express it in a reasonably reduced form.

Starting such a generator may also be a problem. Since such sequences are, in general, far from maximal, there are many possible states for the shift register which are not used in the sequence. In fact, most of the possible states may not be used. A sequence of length 43 requires an 11-stage register so that only 43 of the 2048 possible states are used. To achieve reliable starting, provisions must be made to assure that if the register starts in one of these unused states it will progress to one of the states used in the sequence. The problems of shifting such a sequence are also complex although it appears that they can be solved. In order to have a simpler design procedure (though perhaps at the cost of more equipment), the method described in the following sections was developed.

3. *Generation of a sequence of arbitrary length.* A generator for a linear shift register sequence of maximal length may be considered as a sequential machine and described by a state diagram. The state diagram for the machine shown in Fig. 19 is shown in Fig. 21. Each state is designated by the number on the register. This machine is degenerate in that it has no inputs. Also, in drawing the state diagram it was assumed that the machine included a starter so that the zero state leads to the 16 state. It is a property of a shift register with feedback that a given state can have only one of two possible successors: that produced by feeding back a *one* and that produced by feeding back a *zero*. The feedback selects one of the two possible choices for the successor of each state. If the machine shown in Fig. 21 were altered so that state 11 followed state 22 instead of state 27, a sequence of length 27 would be produced instead of a sequence of length 31. If the alteration in the machine is done properly (Ref. 20), the machine is still self-starting, and since the 1...1, 10...0, and 0...01 words are in the cycle, the sequence may be shifted both right and left. It is a property of the maximal-length linear shift register sequence that it can be shortened by any desired amount; i.e., if a sequence of any arbitrary length is desired, a state can be found whose alternate successor yields a sequence of the desired length.

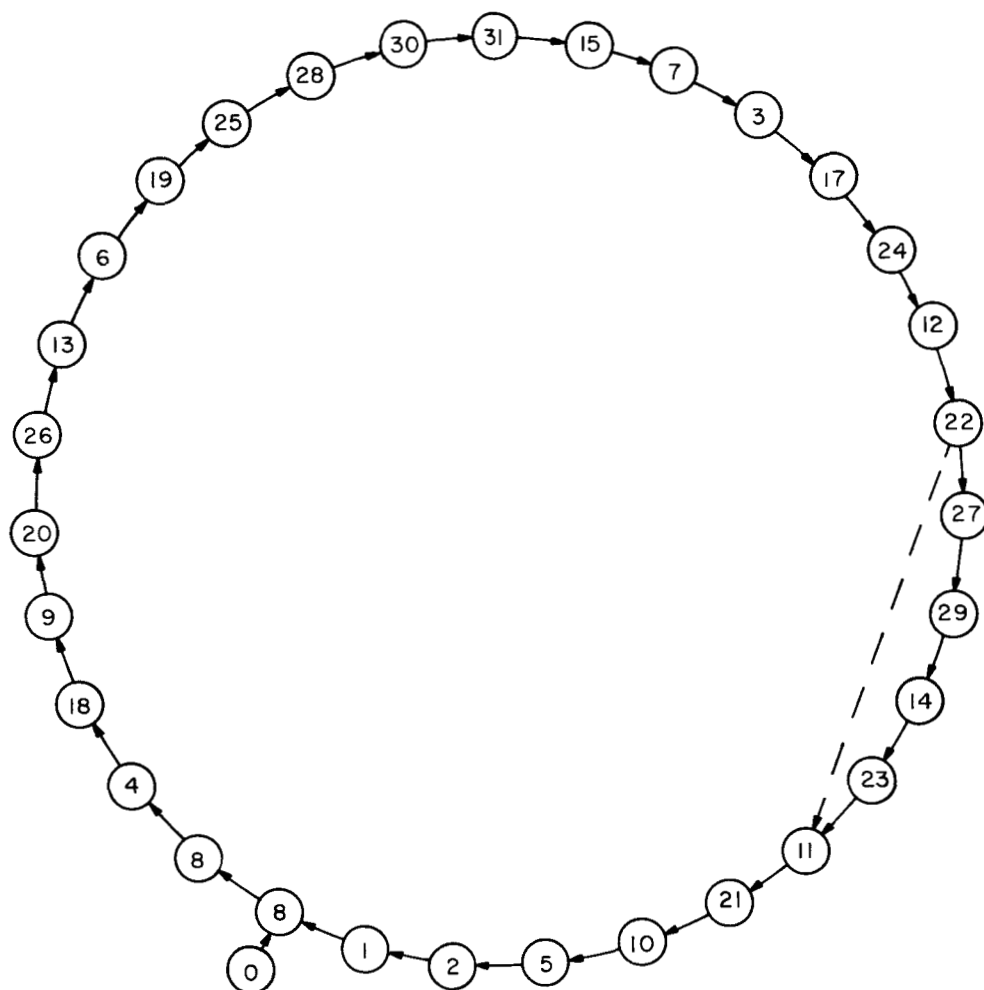


Fig. 21. State diagram for a sequence generator

4. *Indirect generation of other PN sequences.* A PN sequence other than a maximal-length linear shift register sequence can be generated from a sequence of the type described in the previous section. A maximal-length sequence is shortened to the desired length and additional logic is used external to the feedback loop to convert the shortened sequence to the desired PN sequence bit by bit. Such a generator will be self-starting. If the shortened sequence contains the words necessary for shifting, then the shortened sequence can be shifted, and, of course, the derived sequence will be shifted also. If the necessary words are not included in the shortened sequence, then no general approach to the problem of shifting is known. The external modification of a shortened maximal-length sequence is the principal alternative to the method of the previous section for the generation of arbitrary (and especially nonlinear) PN sequences.

5. *Stored sequences.* Instead of *generating* a sequence digit by digit as needed, the entire sequence can be stored in a long shift register, one digit per stage. If the shift register is a delay line, the amount of equipment required for sequences of lengths up to several hundred is not large. The problem of shifting is then handled by precessing the digits in the shift register using standard techniques. While this approach solves the problem of shifting, it begs the question of how the sequence is originally generated.

#### IV. CODED PHASE-COHERENT COMMUNICATIONS

The merits of phase-coherent communications are widely recognized for both discrete and continuous modulation systems (Ref. 21-23). The relative performances of phase-coherent and noncoherent transmission of binary data in the presence of additive white gaussian noise have been analyzed and compared (Ref. 21 and 22). This section considers the result of encoding independent equiprobable binary words or sequences of independent binary digits into sets of binary code words. These code words are transmitted over a channel perturbed by additive white gaussian noise and are detected by being correlated with their stored or locally generated replicas at the receiver.

The word error probabilities and bit error probabilities for low cross-correlation codes are determined as a function of the ratio

$$\frac{\text{received signal energy/bit}}{\text{noise power/unit bandwidth}}$$

The received information rate and the potential channel capacity are also computed. It is shown that in the limit as the code word length and the bandwidth approach infinity, the received information rate approaches the channel capacity for one and only one value of the above ratio.

##### A. The Basic Model

In order to communicate  $n$  bits of information,  $2^n$  distinct choices must be available at the transmitter. These  $2^n$  arbitrary messages or words are to be stored or generated at the transmitter. Depending on the information to be sent, one of the  $2^n$  words is sent over a period of  $nT$  sec,  $T$  being the transmission time allotted per bit. The communication channel is assumed to add an arbitrary disturbance to the transmitted signal (Fig. 22). The ideal receiver computes the conditional probability that each of the possible  $2^n$  words was transmitted over the interval of  $nT$  sec, given the received word. It has been shown by Woodward (Ref. 24), Davies (Ref. 24 and 25), and Fano (Ref. 26) that if the channel disturbance is white gaussian noise, the probability computer consists of  $2^n$  correlators which multiply the incoming signal by each of the  $2^n$  stored or locally generated replicas of the possible transmitted words, integrate over the transmission interval, and are sampled at the end of this time. Thus the output of the  $k$ th correlator, which corresponds to the  $k$ th word  $x_k$  is

$$\int_0^{nT} x_k(t) y(t) dt$$

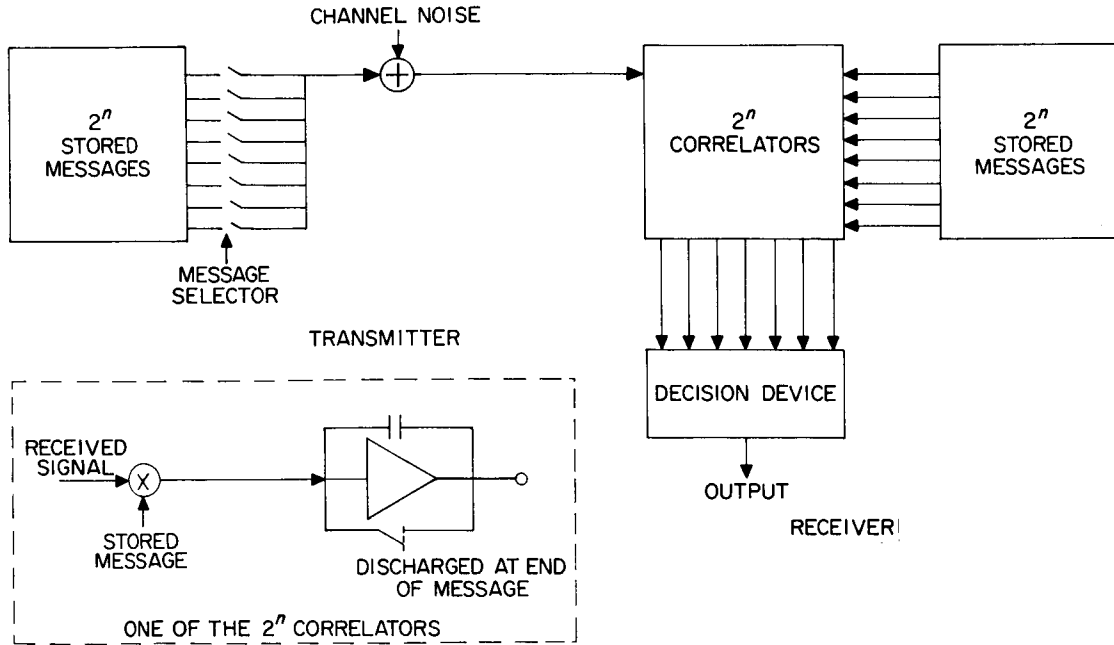


Fig. 22. Basic communications-system model

where  $y(t) = x_m(t) + N(t)$ ,  $x_m(t)$  is the received signal, and  $N(t)$  is the channel noise. If the  $2^n$  words were *a priori* all equally likely to be transmitted with equal energy, i.e.,

$$P(x_i) = P(x_j) \text{ and } \int_0^{nT} x_i^2(t) dt = \int_0^{nT} x_j^2(t) dt$$

for all  $i$  and  $j$ , then the conditional probability that  $x_k$  was sent, given that  $y$  was received, is proportional to the exponential of the output of the  $k$ th correlator (Ref. 26).

$$P(x_k|y) \sim \exp \int_0^{nT} x_k(t) y(t) dt \quad (1)$$

The decision device then examines all the correlator outputs and selects the waveform  $x_k(t)$  corresponding to the maximum correlator output. This is known as maximum-likelihood detection and can be shown to minimize the probability of error when all the signals are equally likely and contain equal energies (Ref. 27).

It follows intuitively that in order to achieve low error probabilities, the waveforms should be as unlike as possible, such that in a noisy channel there will be the least possible chance to make the wrong selection of the word transmitted. More precisely, the cross-correlation coefficients among all pairs of words,

$$\rho = \frac{\int_0^{nT} x_i(t) x_j(t) dt}{\left[ \int_0^{nT} x_i^2(t) dt \int_0^{nT} x_j^2(t) dt \right]^{1/2}} \quad (2)$$

should be made as low as possible. The least possible value of  $\rho$  is  $-1$ . However, this value can be achieved only when the number of words in the set is two ( $n = 1$ ). In this case, if  $x_1(t) = -x_2(t)$ ,  $\rho = -1$ , and the words are said to be antipodal. In general, it is possible to make all the cross-correlation coefficients equal to zero. The set of words is then said to be orthogonal. Actually, it is possible to obtain sets of words for which some or all of the cross-correlations are negative (see Section IV-C).

## B. Realization of the Model

The concepts discussed in the preceding section date back almost ten years. Little has appeared in the literature on the subject of coded phase-coherent communication since that time because of the difficulty in realizing the basic model with stored waveforms (other than for the case of binary waveforms in which one bit at a time is transmitted). The problem is greatly simplified by using binary sequences as the transmitted words, since these can be generated at both transmitter and receiver by relatively simple code generators.

Figure 23 is an example of such a binary coded phase-coherent system. The term "phase-coherent" refers not only to the coherence between the transmitted carrier and the locally generated carrier, but also to that between the transmitted and locally generated code words.

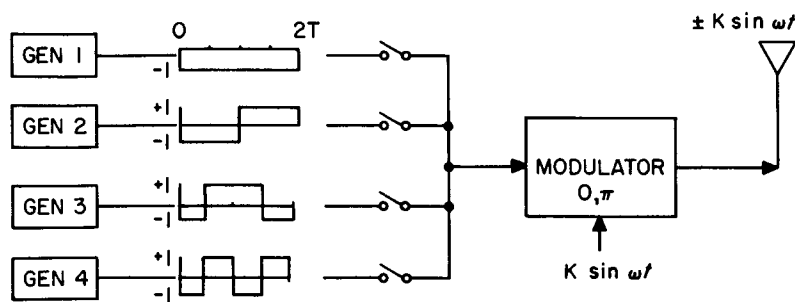
Blocks of two bits of information are transmitted by selecting one of a set of four binary code words. This set is orthogonal, since the words switch between *plus one* and *minus one*, and it is easily verified that

$$\int_0^{2T} x_i(t) x_j(t) dt = 0$$

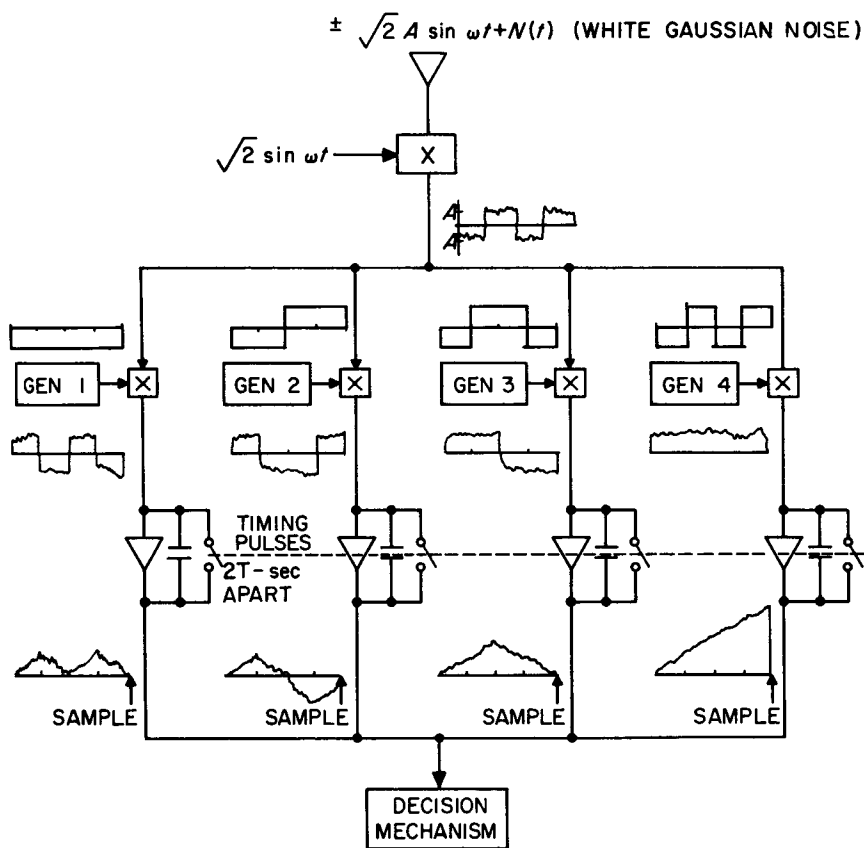
for  $i \neq j$ .

Phase modulation of  $K \sin \omega t$  by  $\pi$  rad when the word is at the  $-1$  level is equivalent to amplitude modulation of the carrier by *plus ones* and *minus ones*. At the receiver, the noisy signal is demodulated and fed to the four correlators. Only the low-frequency component of these inputs is shown in Fig. 23. Actually, the component centered at a frequency of  $2\omega$  rad/sec is eliminated by the integrator, provided  $\omega$  is a multiple of  $\pi/2nT$ , where  $n$  is the number of bits per word.

## CODE GENERATORS



## TRANSMITTER



## RECEIVER

Fig. 23. Binary coded phase-coherent system for transmission of 2 bits/word

Because the code words are orthogonal, the outputs of all correlators, except the one corresponding to the word sent, are zero in the absence of noise. If this were not the case, the noise-free output of the  $i$ th correlator would be proportional to

$$\int_0^{2T} x_i(t) x_j(t) dt$$

when the  $j$ th word was sent.

The properties and generation of the binary code words are discussed in Section IV-C. For the present, other characteristics of the model will be considered. It should be noted that multiplication of the additive noise by the locally generated words does not alter its white gaussian statistics, since multiplying successive uncorrelated samples of noise arbitrarily by *plus one* and *minus one* does not alter the first-order distribution, nor does it render them correlated.

In general, if  $n$  bits are transmitted as one word, the integrating time is  $nT$ . The integrate-and-discharge filter is assumed to produce an attenuation of  $1/nT$ . Thus, the signal will produce an output at time  $nT$  of  $e(nT) = A$ , provided that  $\omega nT$  is a multiple of  $\pi/2$ . The channel noise is white gaussian with spectral density  $N/2B$ . (This input spectral density would produce a power of  $N$  watts at the output of a bandpass filter of bandwidth  $B$ .) The variance at time  $nT$  at the output of the integrate-and-discharge filter is<sup>2</sup>

$$\begin{aligned} \sigma^2 &= E \left[ \frac{1}{(nT)^2} \int_0^{nT} \sqrt{2} \sin \omega t N(t) dt \int_0^{nT} \sqrt{2} \sin \omega u N(u) du \right] \\ &= \frac{1}{(nT)^2} \int_0^{nT} \int_0^{nT} E[N(t) N(u)] 2 \sin \omega t \sin \omega u dt du \end{aligned} \quad (3)$$

Since the noise is white with density  $N/2B$ ,

$$E[N(t) N(u)] = \frac{N}{2B} \delta(t - u)$$

<sup>2</sup>The noise contribution to the input of the integrate-and-discharge filter is  $\sqrt{2N}(t) \sin \omega t$  multiplied by a binary code word. However, since this binary multiplication does not alter the statistics of the noise, it may be neglected.

Therefore,

$$\sigma^2 = \frac{N}{2B(nT)^2} \int_0^{nT} 2 \sin^2 \omega t dt = \frac{N}{2BnT} \quad (4)$$

provided that  $\omega nT$  is a multiple of  $\pi/2$ .

The ratio of peak output signal to the noise standard deviation is

$$\frac{e(nT)}{\sigma} = \frac{A}{(N/2BnT)^{1/2}} = \left( \frac{2A^2nT}{N/B} \right)^{1/2} = \left( \frac{2SnT}{N/B} \right)^{1/2} \quad (5)$$

where  $S = A^2$  is the received signal power. Since  $T$  is the transmission time per bit, the ratio

$$\frac{ST}{N/B} = \frac{\text{received signal energy/bit}}{\text{noise power/unit bandwidth}}$$

This is the basic parameter for communication in the presence of white gaussian noise; the numerator represents the parameters which may be varied by the communicator, while the denominator is the characteristic property of the channel.

In Section IV-C it is shown that if a set of  $2^n$  code words is to be orthogonal each word must contain  $2^n$  symbols; that is, there are  $2^n$  subintervals during which the word may be at either the *plus one* or *minus one* level. Each symbol is of duration  $nT/2^n$  sec. Since the carrier is the sinusoid  $\sin \omega t$ , it is possible to have other sinusoidal carriers at

$$\omega + \frac{2\pi\nu}{nT/2^n}, (\nu = \pm 1, \pm 2, \pm 3, \dots)$$

without interfering with the given signal, provided  $\omega$  is a multiple of  $\pi/(nT/2^n)$ , because over any given subinterval  $nT/2^n$

$$\int_0^{nT/2^n} \sin \omega t \sin \left[ \left( \omega + \frac{2\pi\nu}{nT/2^n} \right) t + \phi \right] dt = 0$$

for  $\nu = \pm 1, \pm 2, \pm 3, \dots$ . Thus, the effective bandwidth occupied by the channel is  $2^n/nT$  cps. If the sinusoidal carrier of the adjacent channel were constrained to alternate between  $\phi = 0$  and  $\phi = \pi$  relative to the given channel (i.e., if it were modulated in the same way), then the adjacent sinusoid could be placed  $\pi/(nT/2^n)$  rad/sec away without interfering, thus making the effective bandwidth occupancy per channel only  $2^{n-1}/nT$  cps.

Another characteristic of orthogonal code sets which is worth noting is that the noise components of the correlator outputs are mutually independent. Of course, the white noise input is the same for each correlator. However, during each of the  $2^n$  code subintervals, the noise will be multiplied by *plus one* or *minus one*. Thus, the cross-correlation between the noise components of any two correlators  $i$  and  $j$  is proportional to

$$\rho_N = E \left[ \sum_{k=0}^{2^n-1} \int_{knT/2^n}^{[(k+1)nT]/2^n} x_i(t) N(t) dt \right] \cdot \left[ \sum_{m=0}^{2^n-1} \int_{mnT/2^n}^{[(m+1)nT]/2^n} x_j(t) N(t) dt \right]$$

where  $x_i(t)$  and  $x_j(t)$  are  $\pm 1$  during any given interval. Since the noise is white, the integral over one interval is independent of the integral over another. Thus,

$$E \left[ \pm \int_{knT/2^n}^{[(k+1)nT]/2^n} N(t) dt \right] \left[ \pm \int_{mnT/2^n}^{[(m+1)nT]/2^n} N(t) dt \right] = 0$$

for  $k \neq m$  and

$$\rho_N = E \left\{ \pm \left[ \int_0^{nT/2^n} N(t) dt \right]^2 \pm \left[ \int_{nT/2^n}^{2nT/2^n} N(t) dt \right]^2 \pm \dots \pm \left[ \int_{nT[1-(1/2^n)]}^{nT} N(t) dt \right]^2 \right\}$$

If the two codes  $x_i$  and  $x_j$  are to be orthogonal, however, there must be exactly as many subintervals during which  $x_i$  and  $x_j$  are of different signs as there are subintervals during which they are of the same sign (see Section IV-C). Thus, for orthogonal codes,  $\rho_N = 0$ .

The optimal decision process and the error probabilities are considered in Section IV-D. The next section will treat some basic properties of binary codes.

### C. Binary Codes

This section contains a brief description of the construction and basic properties of certain error-reducing codes. For a more thorough treatment, the reader is referred to the literature on coding<sup>3</sup> (Ref. 28 and 29).

1. *Orthogonal codes.* A property of a set of orthogonal codes is that the cross-correlation coefficients among all pairs in the set are zero. That is, for the code words

$$\{x_1, x_2, \dots, x_k\} \text{ and } \{y_1, y_2, \dots, y_k\}$$

(where the  $x_i$ 's and  $y_i$ 's can take on the values *plus one* or *minus one*) the sum of the products of corresponding symbols is

$$\sum_{i=1}^k x_i y_i = 0$$

It is sometimes more convenient to write the codes using the symbols *zero* and *one* rather than *plus* or *minus one*. The orthogonal property can then be stated as follows: Two code words are orthogonal if the number of symbol positions in which they are similar equals the number in which they are dissimilar.

Sets of orthogonal codes can be constructed in a multitude of ways, since the  $2^n$  elements of any basis of a  $2^n$ -dimensional vector space over the field of two elements can be made orthogonal to one another (Ref. 30). A simple inductive construction of a set of orthogonal codes follows.

A single bit of information may be sent by selecting from a set of two orthogonal code words of two symbols each:

0 0  
0 1

Two bits might be sent by using the code word set

0 0 0 0  
0 1 0 1  
0 0 1 1  
0 1 1 0

---

<sup>3</sup>It should be noted that these codes are usually classified in the literature as "error-correcting codes" because their redundancy permits correction of up to a given number of erroneous symbols after the message has been received and demodulated on a symbol-by-symbol basis. The present treatment differs from this in that the redundancy is utilized to decode the entire word in one operation rather than piecemeal. Hence, the property which is required of these redundant codes is a uniformly low cross-correlation coefficient.

It should be noted that this set can be constructed by extending the set for one bit both horizontally and vertically. The lower-right-hand square is filled by the complements of these words. A code set for three bits may be constructed by extending the set for two:

0 0 0 0	0 0 0 0
0 1 0 1	0 1 0 1
0 0 1 1	0 0 1 1
0 1 1 0	0 1 1 0
0 0 0 0	1 1 1 1
0 1 0 1	1 0 1 0
0 0 1 1	1 1 0 0
0 1 1 0	1 0 0 1

To prove that the construction yields an orthogonal code set at each step, assume that such a construction exists for  $k$  bits. Then for  $k + 1$  bits, extending the  $2^k$  words vertically yields a set of  $2^{k+1}$  words which are all orthogonal except that each word in the top half is the same as one word in the bottom half. However, extending the word horizontally, the upper half of the extensions is the complement of the lower half. Again, all horizontal extensions are orthogonal, except that each word extension in the top half has as its complement in the bottom half the extension of that word for which the left halves are equal. Thus, each pair of words in the new set has as many similar symbols as it has dissimilar ones. Hence, the set is orthogonal.

2. *Biorthogonal codes.* These codes were first discovered by Muller and Reed (Ref. 28). They can be generated by taking a set of orthogonal code words and adding to it the complements of each word. Thus, biorthogonal codes are really two sets of orthogonal codes which are mutually orthogonal except that each code word in one set has its complement (or antipode) in the other set. A biorthogonal or Reed-Muller code for 4 bits can be constructed from the preceding orthogonal code for 3 bits:

0 0 0 0 0 0 0 0	1 1 1 1 1 1 1 1
0 1 0 1 0 1 0 1	1 0 1 0 1 0 1 0
0 0 1 1 0 0 1 1	1 1 0 0 1 1 0 0
0 1 1 0 0 1 1 0	1 0 0 1 1 0 0 1
0 0 0 0 1 1 1 1	1 1 1 1 0 0 0 0
0 1 0 1 1 0 1 0	1 0 1 0 0 1 0 1
0 0 1 1 1 1 0 0	1 1 0 0 0 0 1 1
0 1 1 0 1 0 0 1	1 0 0 1 0 1 1 0

One advantage of this set over the corresponding orthogonal set is that it requires one-half as many symbols per code word. Thus, the bandwidth required to transmit the same number of bits/sec is cut in half. Also, the average

cross-correlation coefficient among all the codes in a set of  $2^n$  words is  $-1/(2^n - 1)$ , as will now be shown. There are in all  $(2^n - 1) 2^{n-1}$  pairs. The cross-correlations are *minus one* for  $2^{n-1}$  pairs and zero for all the rest. Thus, the average correlation is

$$\frac{(-1) 2^{n-1}}{(2^n - 1) 2^{n-1}} = -\frac{1}{2^n - 1}$$

Sets of biorthogonal codes have equal numbers of *zeros* and *ones*. This is a favorable property since, if all words are equally likely, it assures that the modulating signal will have zero mean; hence, all the power in the carrier will be modulated.

3. *Shift-register codes*. It is known (Ref. 2) that certain shift registers with linear modulo 2 feedback logic produce codes which have two-level autocorrelation functions. If the register has length  $n$  and the code is of maximal length,  $2^n - 1$ , the lower level will be  $-1/(2^n - 1)$  (see Fig. 24). Thus, a set of  $2^n - 1$  codes with a uniform

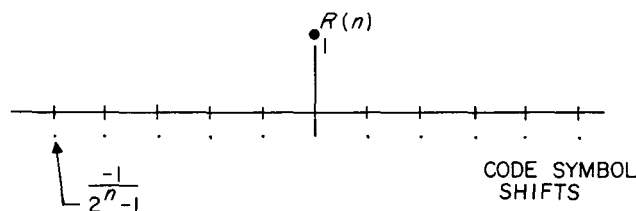


Fig. 24. Autocorrelation function of shift register code

negative cross-correlation coefficient can be constructed by taking all shifted replicas of one maximal-length shift-register sequence. For example, a set of seven code words can be generated by taking all possible shifts of the sequence from a three-stage shift register with linear logic, as shown in Fig. 25. The eighth code word in this figure is the 0-vector (0 0 0 0 0 0 0). The cross-correlation coefficient among all possible pairs is  $-1/(2^n - 1)$ .

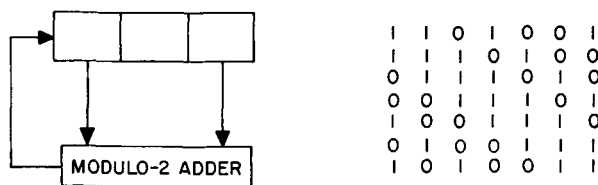


Fig. 25. Shift register and generated code

Shift registers can be used to generate orthogonal or biorthogonal codes quite simply. For example, if a zero is added to every word of the set of Fig. 25 and to the 0-vector, a set of eight orthogonal code words is obtained. By taking the complemented output of the shift register, the complementary orthogonal set is also obtained:

0 1 1 0 1 0 0 1	1 0 0 1 0 1 1 0
0 1 1 1 0 1 0 0	1 0 0 0 1 0 1 1
0 0 1 1 1 0 1 0	1 1 0 0 0 1 0 1
0 0 0 1 1 1 0 1	1 1 1 0 0 0 1 0
0 1 0 0 1 1 1 0	1 0 1 1 0 0 0 1
0 0 1 0 0 1 1 1	1 1 0 1 1 0 0 0
0 1 0 1 0 0 1 1	1 0 1 0 1 1 0 0
0 0 0 0 0 0 0 0	1 1 1 1 1 1 1 1

This example can be generalized to any number of bits.

To demonstrate how elegantly shift-register code generators can be used, consider the case in which the sequence 1001 is to be transmitted by a biorthogonal code sequence. The first digit is transmitted immediately, while the digits 001 are loaded from right to left into the register of Fig. 25, which is made to circulate, and the complemented output digits of the shift register are transmitted. Thus, the whole transmitted sequence is 11100010, one of the words in the above set. If the first digit had been a zero, the uncomplemented output digits of the shift register would have been transmitted. Thus, each possible combination of four binary digits would generate a different word in the set.

As described in Section II, the maximum-length shift-register sequences are merely a subset of the set of all periodic binary sequences with two-level autocorrelation  $R(n)$  satisfying

$$R(n) = \begin{cases} 1 & \text{if } n \equiv 0 \pmod{p} \\ -1/p & \text{if } n \not\equiv 0 \pmod{p} \end{cases}$$

where  $p$  is the period. All the remarks of this section apply to this larger class, except those referring to the special ease of generating linear shift-register sequences.

## D. Optimal Decision and Probability of Error

1. *Orthogonal codes.* The typical receiver for coded phase-coherent communication was shown in Fig. 23. The outputs of the correlators are fed into a device which determines the waveform most probably sent. If the *a priori* probabilities of the various code words are all equal, the disturbance is white gaussian noise, and the energy in all transmitted words is the same, Eq. (1) indicates that the word which was most probably transmitted is that which corresponds to the maximum correlator output.

The probability that the word which was sent will be chosen correctly is equal to the probability that the output of all the other correlators will be smaller than the output of the given correlator. Assume that in the absence of noise, the output of the correlator corresponding to the word sent is  $A$  and that the standard deviation of the output noise of any correlator is  $\sigma$ . For a set of  $2^n$  code words, the probability that the correct one will be chosen is

$$P_c(n) = \int_0^\infty p(x_i) dx_i P(y_1, y_2, \dots, y_j, \dots, y_{2^n-1} < x_i) = \int_0^\infty p(x_i) dx_i \prod_{j=1}^{2^n-1} P(y_j < x_i) \quad (6)$$

where  $p(x_i)$  is the probability density of the output of the correct correlator and

$$P(y_j < x_i) = \int_{-\infty}^{x_i} p(y_j) dy_j$$

is the probability that the output of the  $j$ th incorrect correlator will be less than the correct correlator output. The second equality of Eq. (6) holds because the correlator noise outputs are mutually independent (see Section IV-B). Then, for the given parameters,

$$P_c(n) = \int_{-\infty}^{\infty} \frac{e^{-(x-A)^2/2\sigma^2}}{\sqrt{2\pi}\sigma} dx \left[ \int_{-\infty}^x \frac{e^{-y^2/2\sigma^2}}{\sqrt{2\pi}\sigma} dy \right]^{2^n-1} \quad (7)$$

Making the substitutions  $z = y/\sigma$  and  $u = (x - A)/\sigma$  yields

$$P_c(n) = \int_{-\infty}^{\infty} \frac{e^{-u^2/2}}{\sqrt{2\pi}} du \left[ \int_{-\infty}^{u+(A/\sigma)} \frac{e^{-z^2/2}}{\sqrt{2\pi}} dz \right]^{2^n-1} \quad (8)$$

The probability that a word is in error is

$$P_w(n) = 1 - P_c(n)$$

It was shown in Eq. (5) that the ratio of the output from the correct correlator to the standard deviation of the noise is

$$\frac{A}{\sigma} = \left( \frac{2nST}{N/B} \right)^{1/2}$$

Then,

$$P_w(n) = 1 - \int_{-\infty}^{\infty} \frac{e^{-u^2/2}}{\sqrt{2\pi}} du \left[ \int_{-\infty}^{u + [2nST/(N/B)]^{1/2}} \frac{e^{-z^2/2}}{\sqrt{2\pi}} dz \right]^{2^n - 1} \quad (9)$$

This integral cannot be evaluated analytically in general. However, numerical integration by an IBM 704 computer yielded the results of Fig. 26 for code words containing up to 20 bits of information.

It is also of interest to investigate the behavior of Eq. (9) as  $n$  tends to infinity. Taking limits and using the asymptotic expression for the error function,

$$\begin{aligned} \lim_{n \rightarrow \infty} P_w(n) &= 1 - \lim_{n \rightarrow \infty} \int_{-\infty}^{\infty} \frac{e^{-u^2/2}}{\sqrt{2\pi}} du \left[ 1 - \frac{e^{-nST/(N/B)}}{\sqrt{2\pi} [2nST/(N/B)]^{1/2}} \right]^{2^n - 1} \\ &= 1 - \lim_{n \rightarrow \infty} \left[ 1 - \frac{e^{-nST/(N/B)}}{\sqrt{2\pi} [2nST/(N/B)]^{1/2}} \right]^{2^n} \end{aligned}$$

To evaluate this limit, consider the limit of its logarithm:

$$\lim_{n \rightarrow \infty} \ln \left[ 1 - \frac{e^{-n\xi}}{\sqrt{2\pi} (2n\xi)^{1/2}} \right]^{2^n} = \lim_{n \rightarrow \infty} 2^n \ln \left[ 1 - \frac{e^{-n\xi}}{\sqrt{2\pi} (2n\xi)^{1/2}} \right]$$

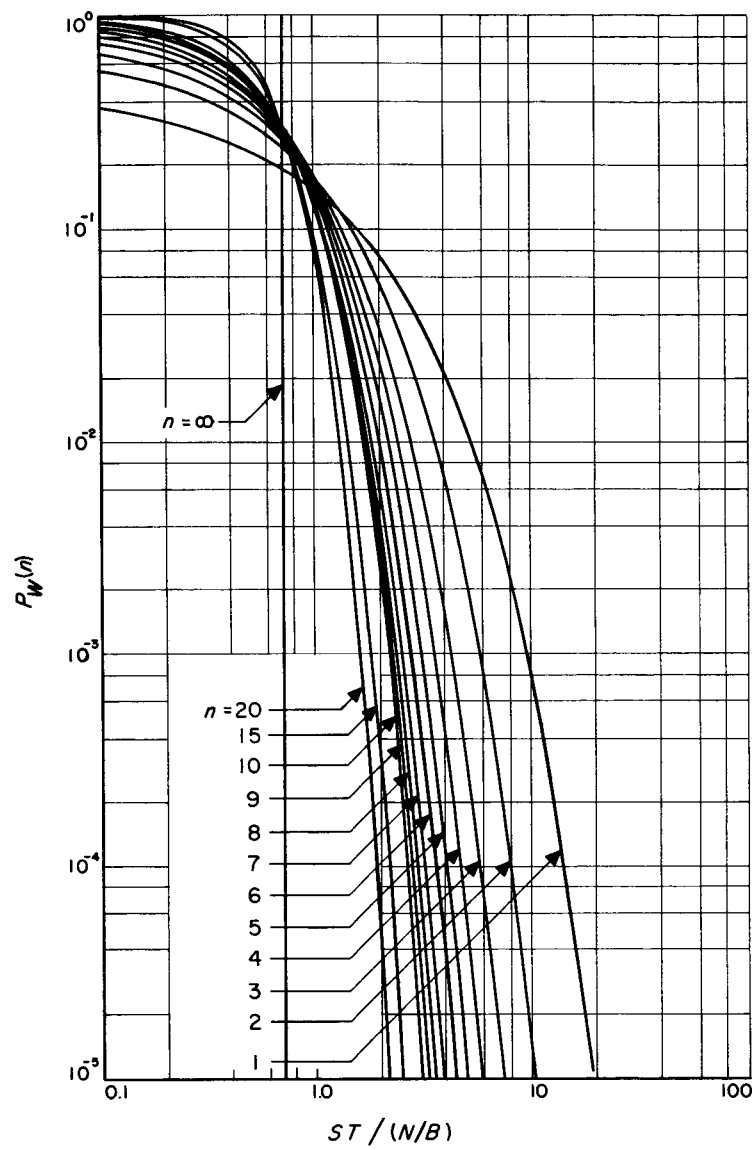


Fig. 26. Word error probability—orthogonal codes

where

$$\xi = \frac{ST}{N/B}$$

Treating  $n$  as a continuous variable and using l'Hospital's Rule,

$$\begin{aligned} \lim_{n \rightarrow \infty} \frac{\ln \left[ 1 - \frac{e^{-n\xi}}{\sqrt{2\pi} (2n\xi)^{1/2}} \right]}{2^{-n}} &= \lim_{n \rightarrow \infty} \frac{\left[ 1 - \frac{e^{-n\xi}}{2(\pi n\xi)^{1/2}} \right]^{-1} \left[ \frac{e^{-n\xi} (2n^{1/2} \xi^{3/2} + n^{-1/2} \xi^{1/2})}{4\sqrt{\pi} n\xi} \right]}{-2^{-n} \ln 2} \\ &= \lim_{n \rightarrow \infty} \frac{-(2e^{-\xi})^n \xi^{1/2}}{2(\pi n)^{1/2} \ln 2} \end{aligned}$$

If

$$2e^{-\xi} < 1 \quad \text{or} \quad \xi = \frac{ST}{N/B} > \ln 2$$

this limit of the logarithm is zero; otherwise, it is unbounded negatively. Thus, for

$$\frac{ST}{N/B} > \ln 2, \quad \lim_{n \rightarrow \infty} P_w(n) = 1 - e^0 = 0 \quad (10)$$

while for

$$\frac{ST}{N/B} \leq \ln 2, \quad \lim_{n \rightarrow \infty} P_w(n) = 1 - e^{-\infty} = 1 \quad (11)$$

that is, the error probability for an infinitely long word jumps from one to zero at the critical value

$$\frac{\text{received signal energy/bit}}{\text{noise power/unit bandwidth}} = \ln 2$$

2. *Biorthogonal codes.* To demodulate a set of  $2^n$  biorthogonal code words carrying  $n$  bits, only  $2^{n-1}$  correlators are required. This is due to the fact that in the absence of noise any one correlator will produce a positive voltage  $+A$  at time  $nT$  for one code word, a negative voltage  $-A$  for its complement, and zero voltage for all the rest. Thus, only one orthogonal code set need be generated at the receiver. The first step in the decision process is to establish whether the voltage at time  $nT$  at the output of a given correlator is positive or negative; thereafter, the

situation is the same as for orthogonal codes, and the optimal decision in the presence of white gaussian noise is to choose the one corresponding to the greatest output.

The correct word will be selected if the absolute values of the outputs of all the other correlators are less than that of the given one, and if the output of the correct correlator is of the right sign. Without loss of generality, assume that a word has been sent which produces a voltage  $+A$  at time  $nT$  on correlator  $x$ . The probability that it will be selected by the decision process is

$$P_c(n) = \int_0^\infty p(x) dx \left[ \prod_{j=1}^{2^{n-1}-1} P(|y_j| < |x|) \right]$$

where

$$P(|y_j| < |x|) = \int_{-x}^x p(y_j) dy_j$$

(This expression is valid because the noise outputs of the correlators are independent since the noise components are again multiplied by orthogonal words.)

In terms of the gaussian densities,

$$P_c(n) = \int_0^\infty \frac{e^{-(x-A)^2/2\sigma^2}}{\sqrt{2\pi}\sigma} dx \left[ \int_{-x}^x \frac{e^{-y^2/2\sigma^2}}{\sqrt{2\pi}\sigma} dy \right]^{2^{n-1}-1}$$

Making the substitutions,

$$v = \frac{x-A}{\sigma} \quad \text{and} \quad z = \frac{y}{\sigma}$$

and recalling from Eq. (5) that

$$\frac{A}{\sigma} = \left( \frac{2nST}{N/B} \right)^{1/2}$$

the word error probability for biorthogonal coding is

$$P_w(n) = 1 - P_c(n) = 1 - \int_{-\left[2nST/(N/B)\right]^{1/2}}^{\infty} \frac{e^{-v^2/2}}{\sqrt{2\pi}} dv \left[ \int_{-\left\{v + \left[2nST/(N/B)\right]^{1/2}\right\}}^{v + \left[2nST/(N/B)\right]^{1/2}} \frac{e^{-z^2/2}}{\sqrt{2\pi}} dz \right]^{2^{n-1}-1} \quad (12)$$

This expression was also evaluated, using an IBM 704, for various values of  $n$  and the results plotted in Fig. 27. Its limit as  $n$  approaches infinity can be computed in the same way as for orthogonal codes, and the result is the same.

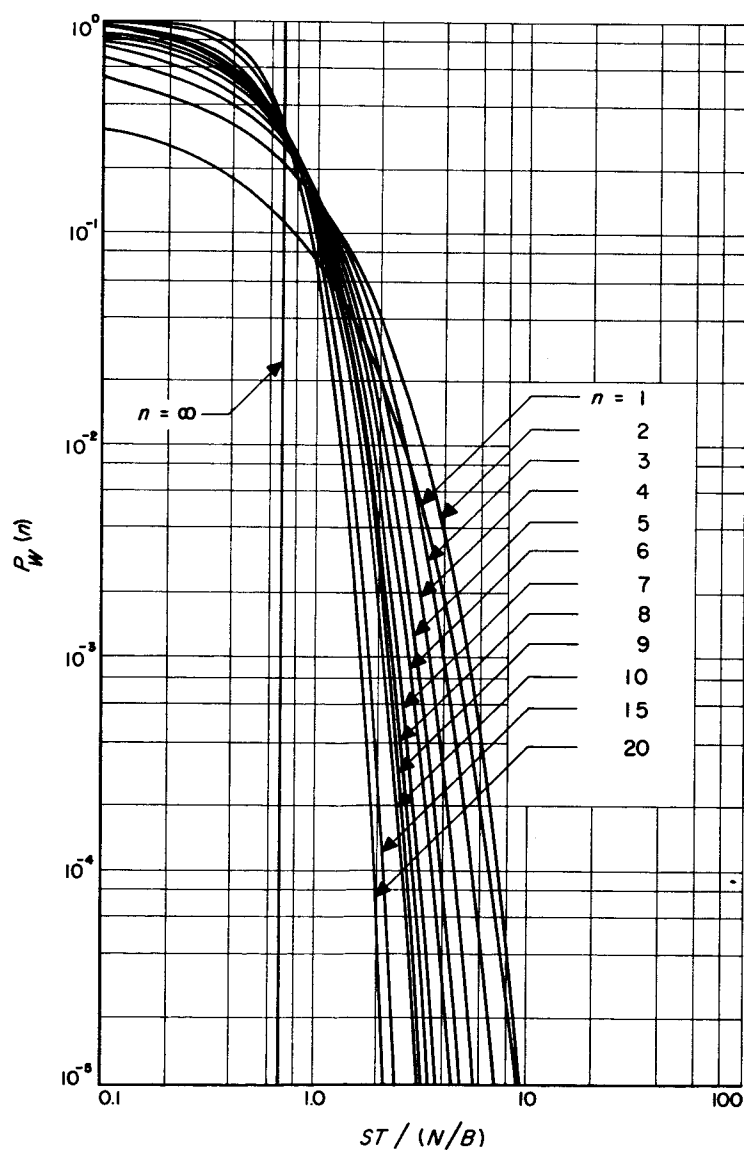


Fig. 27. Word error probability - biorthogonal codes

3. *Comparison of coded and uncoded word error probabilities.* If a single bit were to be sent using a biorthogonal code, the code set would degenerate to two words of one symbol each. This is the special case of communication with two antipodal signals (such as *plus one* and *minus one*). In this situation, which is referred to as uncoded, the probability that each bit is in error is obtained by letting  $n = 1$  in Eq. (12):

$$P_B = 1 - \int_{-[2ST/(N/B)]^{1/2}}^{\infty} \frac{e^{-v^2/2}}{\sqrt{2\pi}} dv = \int_{-\infty}^{-[2ST/(N/B)]^{1/2}} \frac{e^{-v^2/2}}{\sqrt{2\pi}} dv \quad (13)$$

If it is desired to transmit an  $n$ -bit word by sending one bit at a time by means of antipodal signals, the probability that the word will be received in error is one minus the product of the probabilities that each bit will be detected correctly. Thus,

$$P_w(n) = 1 - (1 - P_B)^n \quad (14)$$

This expression is plotted in Fig. 28. For the sake of comparison, Fig. 29 and 30 show the word error probabilities for coded and uncoded transmission, and, as might be expected, the two coding schemes produce almost identical results for large  $2^n$ . Also, the improvement due to coding for  $n = 10$  is almost twice as great as for  $n = 5$ .

### E. Bit Error Probabilities

The significant measure of a communication system's performance depends upon its use. If a sequence of  $n$ -bit messages such as teletype or sampled data is to be sent, the word error probability is the important parameter. On the other hand, if a sequence of independent bits is sent, the bit error probability should be determined.

For orthogonal coding, since all errors are equally probable, the expected number of bits in error when  $n$ -bit coded word has been detected incorrectly is

$$\frac{\sum_{i=1}^n i \binom{n}{i}}{\sum_{i=1}^n \binom{n}{i}} = \frac{n 2^{n-1}}{2^n - 1}$$

Thus, the conditional probability that a given bit is in error when the  $n$ -bit word within which it was encoded is incorrect is  $2^{n-1}/(2^n - 1)$ .

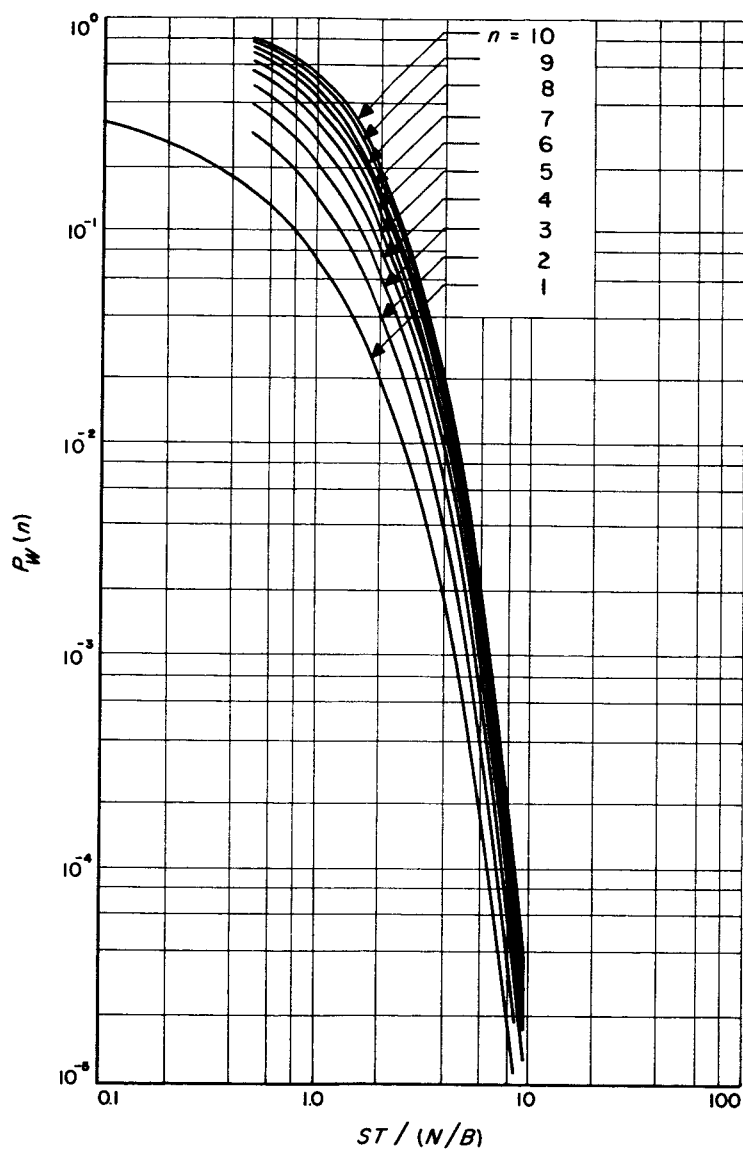


Fig. 28. Word error probability -uncoded

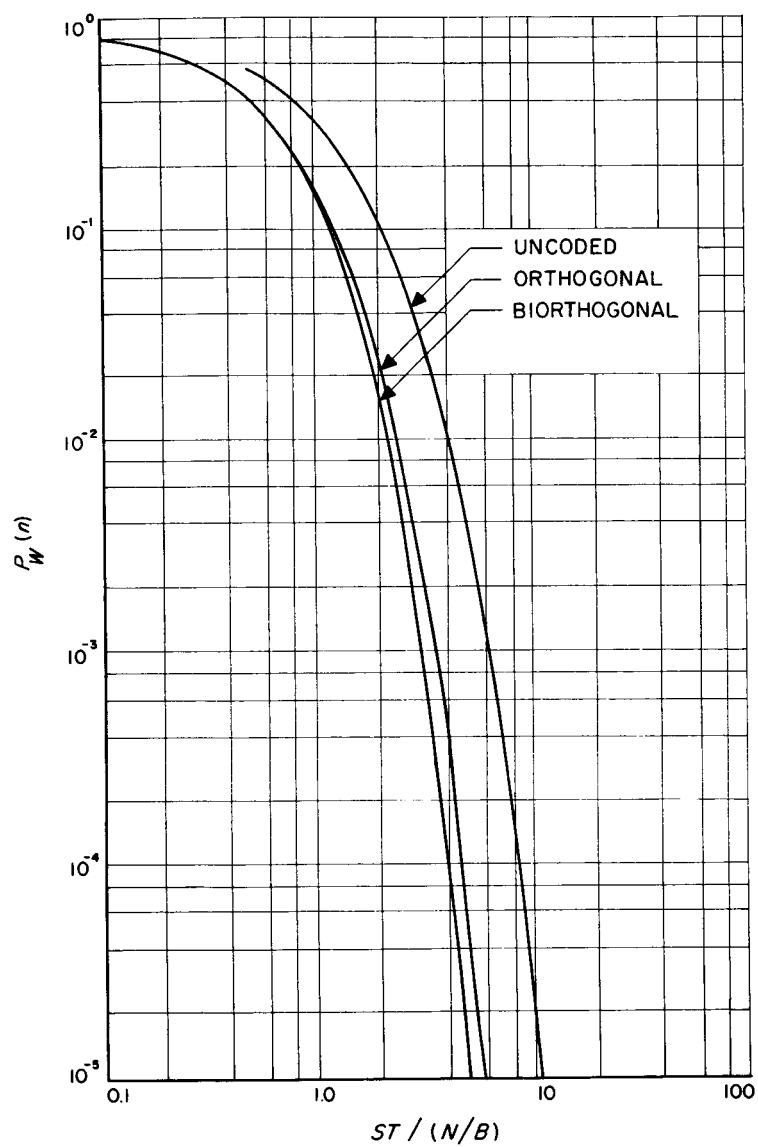


Fig. 29. Comparison of coded and uncoded word error probabilities;  $n = 5$

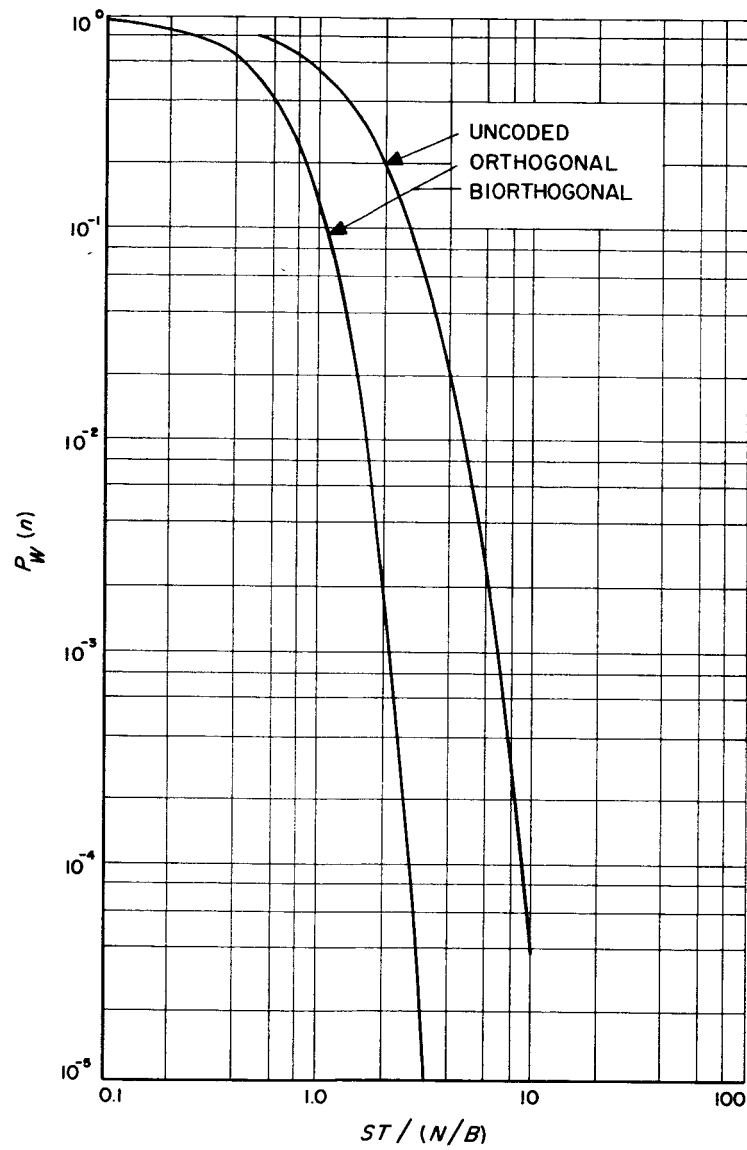


Fig. 30. Comparison of coded and uncoded word error probabilities;  $n = 10$

Then, in terms of the word error probability  $P_w(n)$  for an  $n$ -bit orthogonal code word, the bit error probability is

$$P_B(n) = \frac{2^{n-1}}{2^n - 1} P_w(n) \quad (15)$$

Figure 31 presents these probabilities for orthogonal codes.

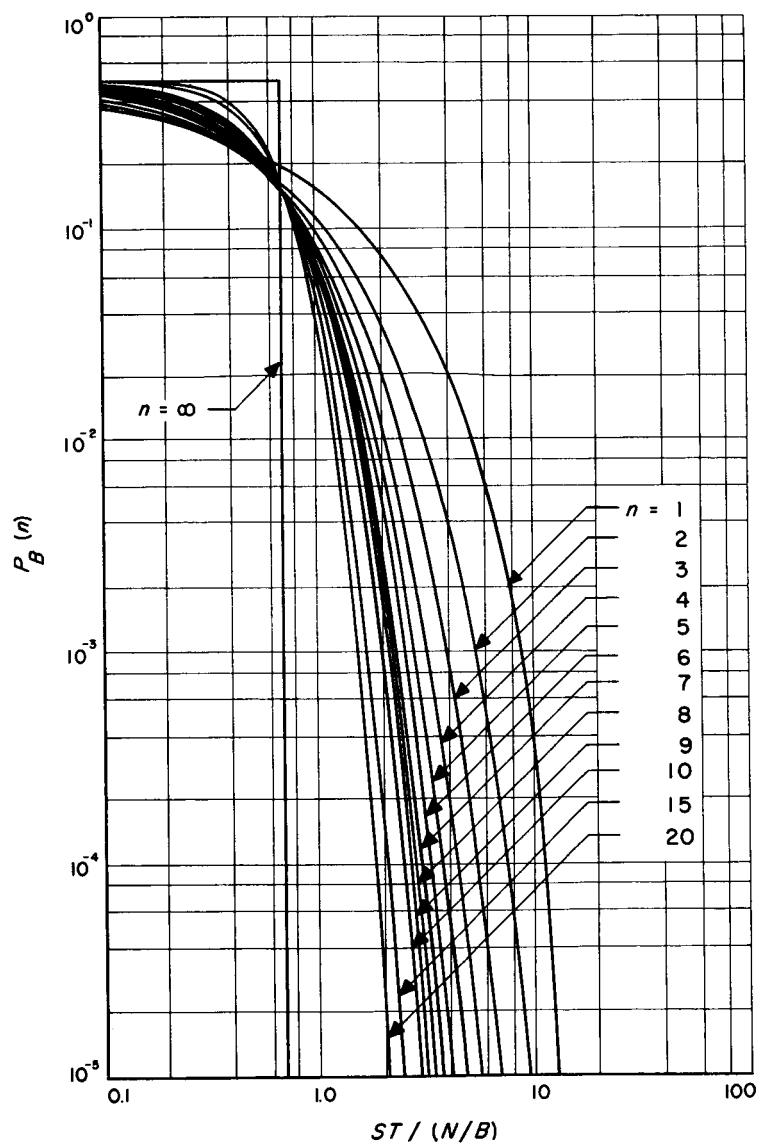


Fig. 31. Bit error probability—orthogonal codes

For biorthogonal codes, the situation is somewhat more complicated. The probability of selecting the code word antipodal or complementary to the transmitted word is much lower than that of selecting a word orthogonal to it. Following the derivation of Section IV-D, this probability, which shall be termed an error of the first kind, is

$$P_1(n) = \int_{-\infty}^{-[2nST/(N/B)]^{1/2}} \frac{e^{-v^2/2}}{\sqrt{2\pi}} dv \left[ \int_{-\{v+[2nST/(N/B)]^{1/2}\}}^{v+[2nST/(N/B)]^{1/2}} \frac{e^{-z^2/2}}{\sqrt{2\pi}} dz \right]^{2^{n-1}-1} \quad (16)$$

The probability of selecting one of the  $2^n - 2$  code words orthogonal to the transmitted word, which shall be termed an error of the second kind, is the total probability of error [ $P_w(n)$  of Eq. (12)] less  $P_1(n)$ .

$$P_2(n) = P_w(n) - P_1(n) \quad (17)$$

It is assumed that complementary message words are coded into complementary code words so as to minimize the probability that a word error will cause all bits to be in error. Then, if an error of the first kind is made, the number of bits in error is exactly  $n$ ; the conditional bit error probability, given that an error of the first kind was made, is 1. If an error of the second kind is made, the expected number of bits in error is

$$\frac{\sum_{i=1}^{n-1} i \binom{n}{i}}{\sum_{i=1}^{n-1} \binom{n}{i}} = \frac{(n-1)2^{n-2}}{2^{n-1}-1}$$

Thus, the bit error probability, given that an error of the second kind was made, is

$$\frac{(n-1)2^{n-2}}{n(2^{n-1}-1)}$$

The total bit error probability for biorthogonal codes is then

$$P_B(n) = P_1(n) + \frac{(n-1)2^{n-2}}{n(2^{n-1}-1)} P_2(n) \quad (18)$$

Figure 32 represents these results.

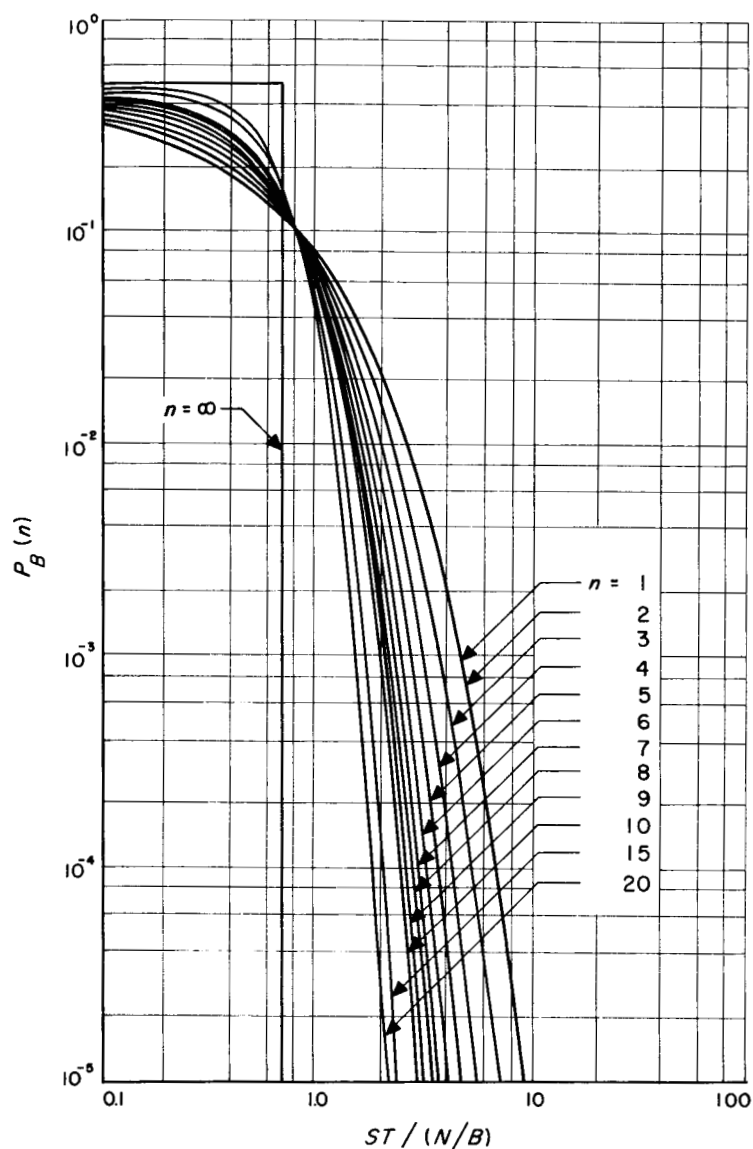


Fig. 32. Bit error probability - biorthogonal codes

#### F. Information Rate and Channel Capacity

A measure of communication-system performance which transcends the subjective use of the received message is the channel information rate. Naturally, in the absence of noise, the information rate  $H$  is equal to the transmission rate,  $1/T$  bits/sec. A noisy channel, however, increases the uncertainty of the received information and hence decreases the rate of actual information received. This uncertainty or decrease in information, treated in the context of the definitions of information theory for a discrete channel (Ref. 31), has been shown to be

$$H_x(y) = -\sum_i \sum_j P(x_i, y_j) \log_2 P(y_j | x_i) \quad (19)$$

where  $x_i$  and  $y_i$  are arbitrary transmitted and received signals, respectively;  $H_x(y)$  is the uncertainty that  $y$  was received when  $x$  was sent;  $p(x_i, y_j)$  is the joint probability that  $x_i$  was sent and  $y_j$  received; and  $p(y_j | x_i)$  is the conditional probability that  $y_j$  was received, given that  $x_i$  was sent.

1. *Orthogonal codes.* The transition probability diagram between transmitted and received words for codes having zero cross-correlation coefficients is shown in Fig. 33. Only a portion of the diagram need be shown, since

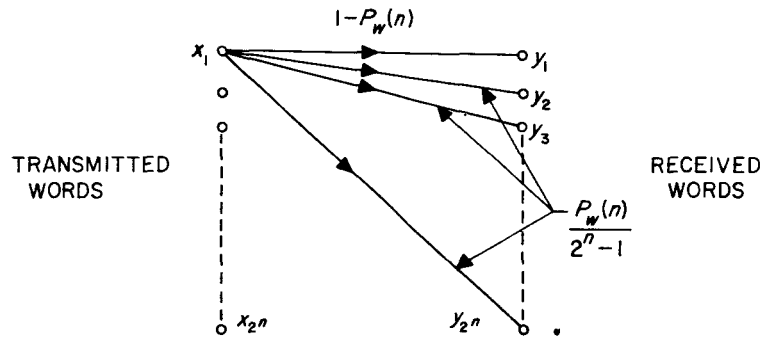


Fig. 33. Diagram of transition probabilities in the presence of noise—orthogonal codes

the pattern is repetitive. The probability that the transmitted word was received correctly is  $1 - P_w(n)$ , while the probability that any one of the other  $2^n - 1$  words was incorrectly chosen is  $[P_w(n)] / (2^n - 1)$ . Applying Bayes' Rule to Eq. (19),

$$H_x(y) = -\sum_i P(x_i) \sum_j P(y_j | x_i) \log_2 P(y_j | x_i) = -\sum_j P(y_j | x) \log_2 P(y_j | x)$$

since the errors are independent of the words sent. Then, with the transition probabilities of Fig. 33,

$$H_x(y) = -[1 - P_w(n)] \log_2 [1 - P_w(n)] - P_w(n) [\log_2 P_w(n) - \log_2 (2^n - 1)] \quad (20)$$

is the equivocation per  $n$ -bit word.

Since the rate of transmission is  $1/T$  bits/sec or  $1/nT$  words/sec, the equivocation rate is  $[H_x(y)] / nT$ . Subtracting this from the transmission rate<sup>4</sup> yields the received information rate:

<sup>4</sup>In general, the equivocation  $H_x(y)$  should be subtracted from the received entropy  $H(y) = \sum_i P(y_i) \log P(y_i)$ . However, in this case, since the transition probabilities due to noise are all the same,  $[H(y)] / nT = [H(x)] / nT = 1/T$ , the transmission rate.

$$H = \frac{1}{T} \left\{ 1 + \frac{[1 - P_w(n)] \log_2 [1 - P_w(n)] + P_w(n) [\log_2 P_w(n) - \log_2 (2^n - 1)]}{n} \right\} \frac{\text{bits}}{\text{sec}} \quad (21)$$

This measure is plotted in Fig. 34 as a function of the basic parameter  $ST/(N/B)$ . It is seen that as  $n$  increases, the speed with which the information rate approaches  $1/T$  increases. In the limit, as  $n$  approaches infinity,  $P_w(n)$  was shown to go from 1 to 0 stepwise at  $ST/(N/B) = \ln 2$ . Thus, the information rate also behaves stepwise in the limit going from 0 to  $1/T$  at this value of the parameter.

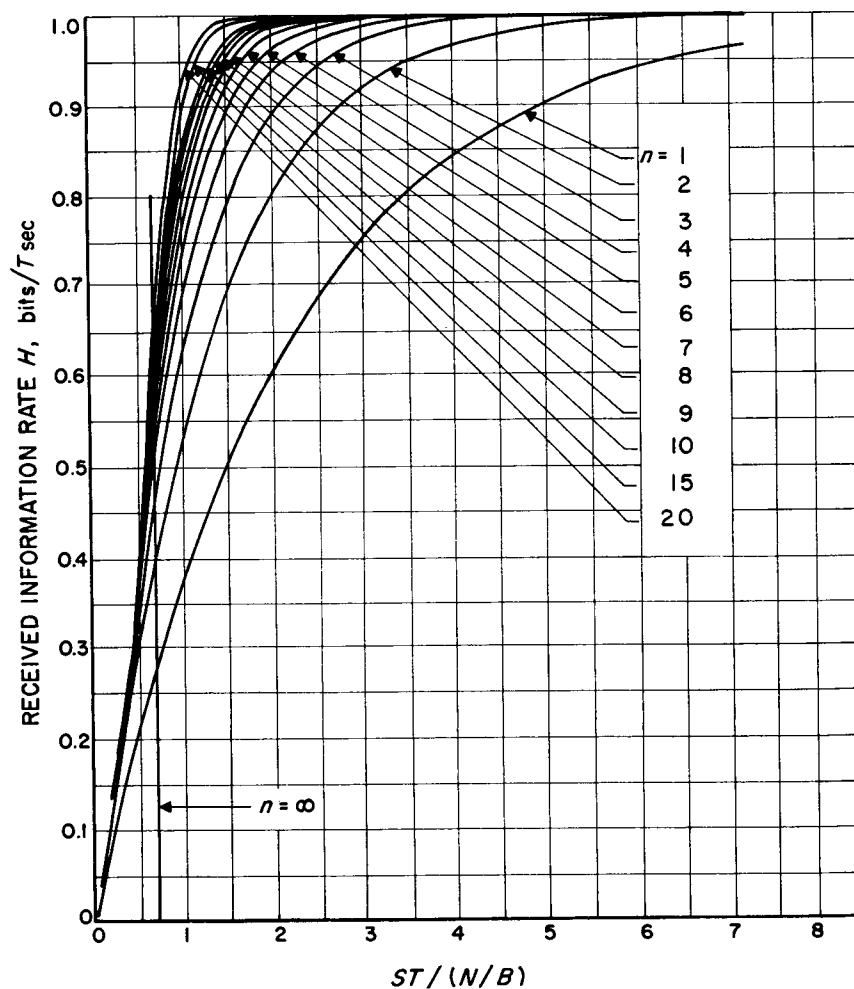


Fig. 34. Received information rate—orthogonal codes

Another important measure of information theory is the celebrated channel capacity. For our purposes, this may be defined as

$$C \left( \frac{ST}{N/B} \right) = \max_{\text{all possible coding methods}} \left[ H \left( \frac{ST}{N/B} \right) \right] \quad (22)$$

Shannon (Ref. 32) has shown that this maximum can be achieved for continuous gaussian-distributed signals and white gaussian noise in the limit as the number of bits per message becomes infinite. It is given by the well-known formula,

$$C = B \log_2 \left( 1 + \frac{S}{N} \right)$$

As was shown in Section IV-C, the bandwidth occupancy of orthogonal codes is  $B = (2^n/nT)$  cps. Thus,

$$\frac{S}{N} = \frac{ST}{N/B} \left( \frac{n}{2^n} \right)$$

and

$$C = \frac{1}{T} \left\{ \frac{2^n}{n} \log_2 \left[ 1 + \frac{n}{2^n} \left( \frac{ST}{N/B} \right) \right] \right\} \frac{\text{bits}}{\text{sec}} \quad (23)$$

In the limit, the capacity behaves as

$$\lim_{n \rightarrow \infty} C = \frac{1}{T} \left( \frac{ST}{N/B} \right) \times \lim_{n \rightarrow \infty} \log_2 \left[ 1 + \frac{n}{2^n} \left( \frac{ST}{N/B} \right) \right]^{[2^n/n] [(N/B)/ST]} = \frac{1}{T \ln 2} \left( \frac{ST}{N/B} \right) \frac{\text{bits}}{\text{sec}} \quad (24)$$

Channel capacity is plotted in Fig. 35 as a function of  $ST/(N/B)$  for several values of  $n$ .

It is of interest to determine how near to the absolute maximum an information rate can be achieved with a coded phase-coherent communication system. As has been noted, in the limit as the message length and bandwidth go to infinity, an information rate of  $1/T$  can be achieved with  $ST/(N/B) = \ln 2$ . Equation (24) shows that for this value of  $ST/(N/B)$ , the channel capacity is, in fact,  $1/T$ ; thus, in the limit of infinite coding, the channel capacity

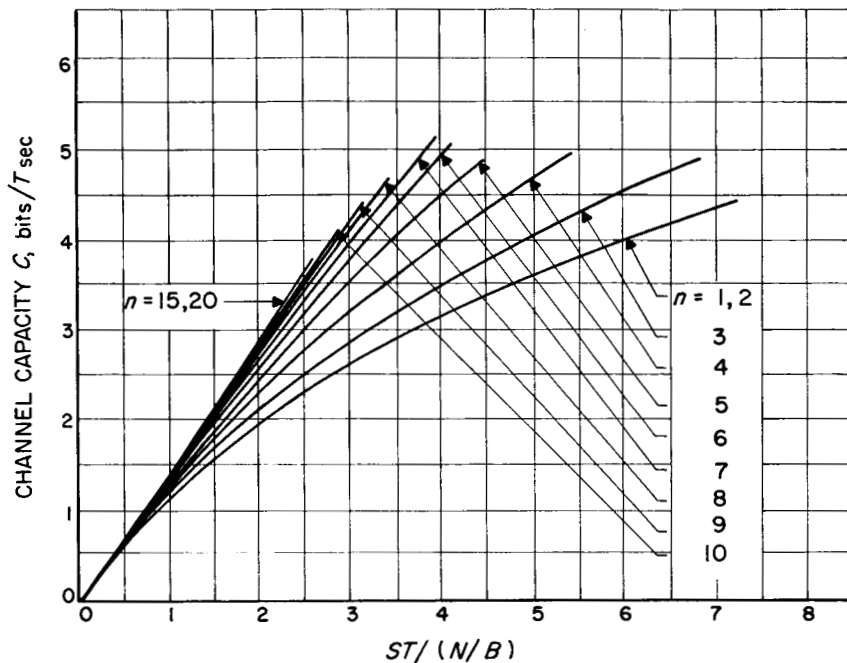


Fig. 35. Channel capacity - orthogonal codes

can be achieved. For higher values of  $ST/(N/B)$ ,  $\lim_{n \rightarrow \infty} H$  naturally remains constant at  $1/T$  while  $\lim_{n \rightarrow \infty} C$  increases linearly; thus, the efficiency  $H/C$  decreases in inverse proportion to the parameter (Fig. 36).

Two observations with regard to these parameters are in order. When bandwidth is at a premium, channel capacity and channel efficiency are important parameters. However, when bandwidth is of secondary importance and the basic purpose is only to transmit with as low an error probability or as high an information rate ( $H$ ) as possible, the channel efficiency is not a significant measure of performance. Also, when reasonably error-free reception is required, it is not meaningful to speak of the information rate or channel efficiency. For example, it is seen in Fig. 34 that the received information rate is 82% of the transmission rate for a 10-bit word and a ratio of  $ST/(N/B) = 1$ . However, from Fig. 26 and 31, it is seen that the word error probability is 0.12 and the bit error probability is 0.06 for this case, which indicates rather poor reception.

2. *Biorthogonal codes.* The transition probabilities for this type of coding were discussed in Section IV-E. The transition diagram is shown in Fig. 37.

$$\begin{aligned}
 H_x(y) &= - \sum_j P(y_j | x_i) \log_2 P(y_j | x_i) \\
 &= - [1 - P_w(n)] \log_2 [1 - P_w(n)] - P_2(n) [\log_2 P_2(n) - \log_2 (2^n - 2)] - P_1(n) \log_2 P_1(n) \quad (25)
 \end{aligned}$$

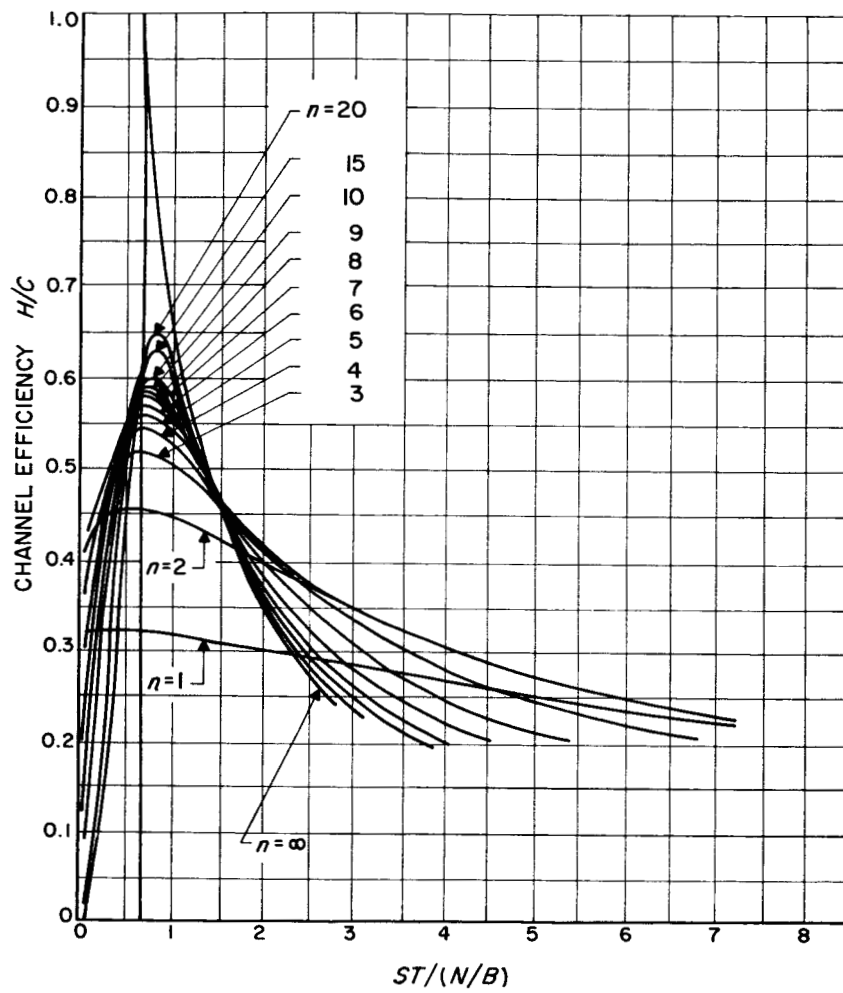


Fig. 36. Channel efficiency—orthogonal codes

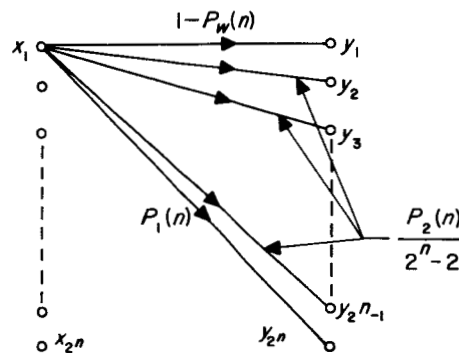


Fig. 37. Diagram of transition probabilities in the presence of noise—biorthogonal codes

and

$$H = \frac{1}{T} - \frac{H_x(y)}{nT} \frac{\text{bits}}{\text{sec}} \quad (26)$$

The bandwidth occupancy for biorthogonal codes is only  $2^{n-1}/nT$ , and half that for orthogonal codes. Thus,

$$C = \frac{1}{T} \left\{ \frac{2^{n-1}}{n} \log_2 \left[ 1 + \frac{n}{2^{n-1}} \left( \frac{ST}{N/B} \right) \right] \right\} \frac{\text{bits}}{\text{sec}} \quad (27)$$

and

$$\lim_{n \rightarrow \infty} C = \frac{1}{T \ln 2} \left( \frac{ST}{N/B} \right) \frac{\text{bits}}{\text{sec}}$$

Equations (26) and (27) and their ratio are plotted in Fig. 38, 39, and 40, respectively. It is seen that the efficiency for small  $n$  is greater than for orthogonal codes because of the lesser bandwidth occupancy. However, for large  $n$ , the received information rate and channel capacity, and hence the efficiency, are about the same for both types of coding.

## G. Conclusions

Coding of information into sets of sequences characterized by low cross-correlation coefficients has the effect of reducing the error probabilities at the cost of expanding the bandwidth for a fixed rate of transmission. If the time allotted per bit is  $T$  sec and the number of bits per code word is  $n$ , the transmission rate is  $1/T$  bits/sec or  $1/nT$  words/sec, and the effective bandwidth is  $2^n/nT$  cps for orthogonal codes and  $2^{n-1}/nT$  cps for biorthogonal codes.

If five bits of information are to be sent with a word error probability of  $10^{-3}$ , the use of a biorthogonal code word will reduce the required

$$\frac{\text{received signal energy/bit}}{\text{noise power/unit bandwidth}}$$

ratio by 3 db under that required for similar performance with bit-by-bit detection. If ten bits are to be sent with the same word error probability, biorthogonal coding reduces the ratio required without coding by 5 db. Orthogonal codes are very nearly as effective as biorthogonal codes for  $n$  greater than 5, but require twice as much bandwidth.

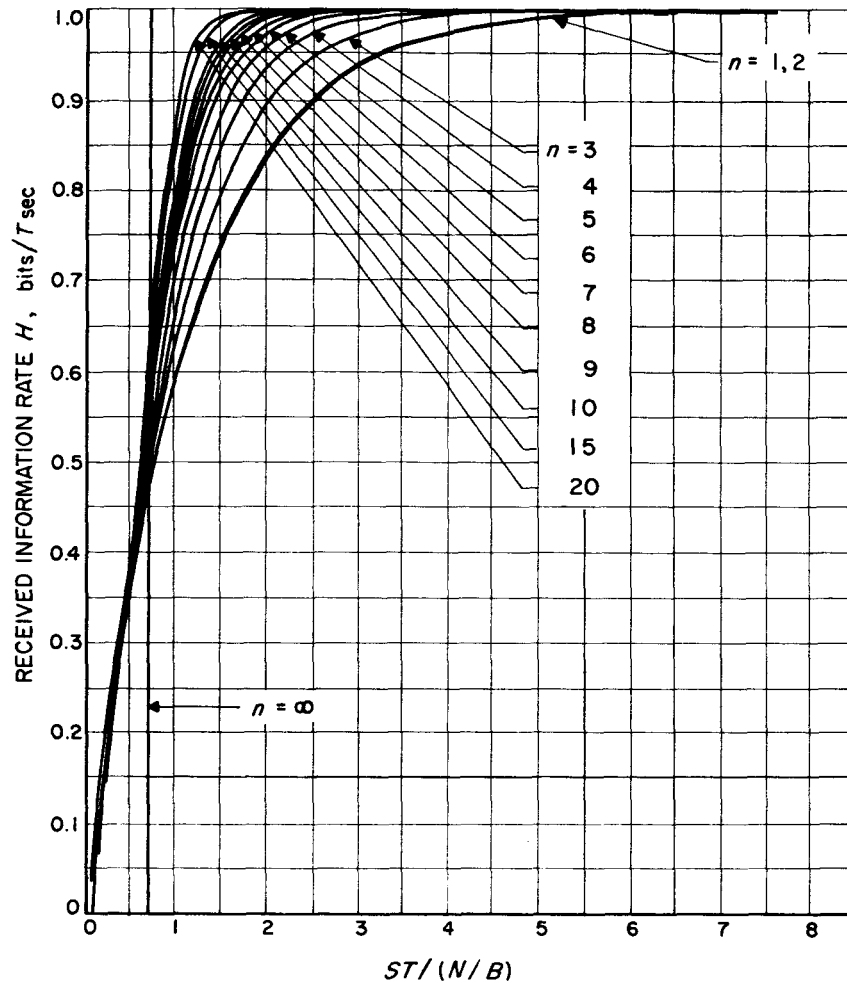


Fig. 38. Received information rate—biorthogonal codes

In the limit as the number of bits per code word and the bandwidth approach infinity, the error probability approaches zero for a

$$\frac{\text{received signal energy/bit}}{\text{noise power/unit bandwidth}}$$

ratio greater than  $\ln 2$ , but it approaches one when the ratio is less than or equal to  $\ln 2$ . Consequently, the received information rate goes stepwise from 0 to  $1/T$  bits/sec at this value of the ratio. In the limit, the channel capacity is a linear function of the above ratio. The

$$\frac{\text{received information rate}}{\text{channel capacity}}$$

or channel efficiency, is shown to approach *one* only when the ratio is  $\ln 2$ . For lower values, the efficiency approaches zero, while for higher values the asymptotic behavior is inversely proportional to the ratio.

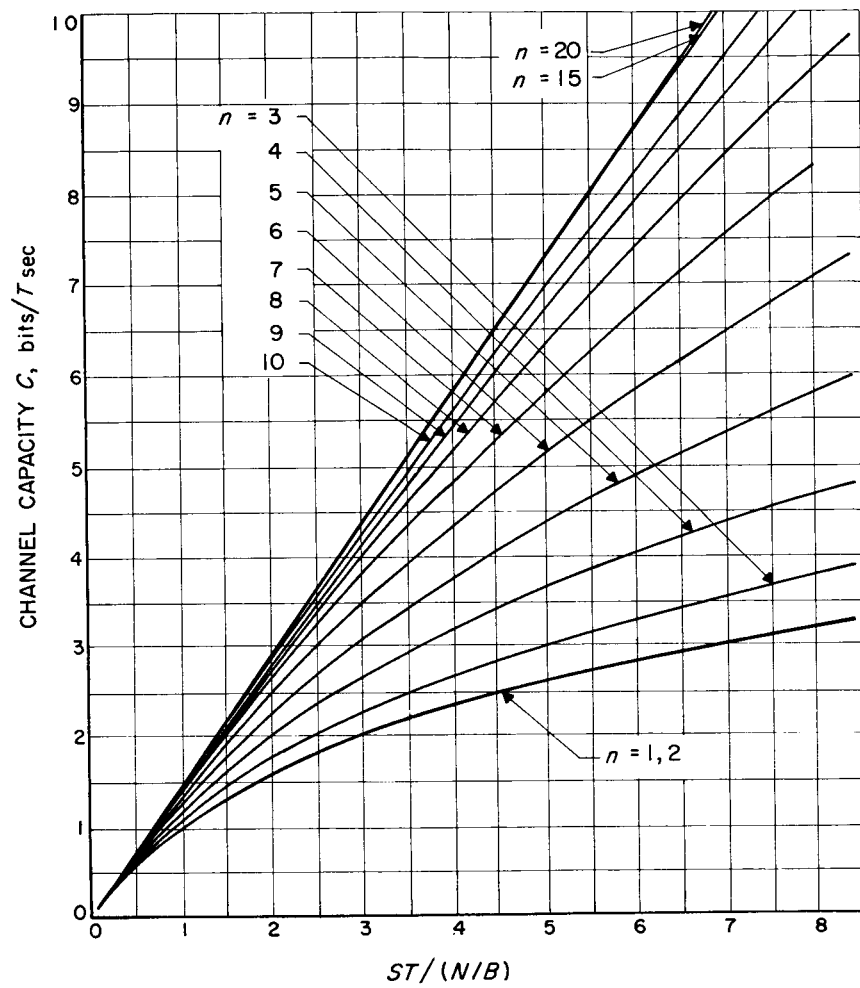


Fig. 39. Channel capacity—biorthogonal codes

## REFERENCES

1. Reed, I. S., "A Class of Multiple-Error-Correcting Codes and the Decoding Scheme," *IRE Transactions on Information Theory*, PGIT-4, September 1954, pp. 38 - 49.
2. Golomb, S. W., *Sequences with Randomness Properties*, Terminal Progress Report under Contract Req. No. 639498, Glenn L. Martin Co., Baltimore, Md., June 1955.
3. Zierler, N., "Linear Recurring Sequences," *Journal of the Society for Industrial and Applied Mathematics*, Vol. 7, 1959, pp. 31 - 48.
4. Paley, R. E. A. C., "On Orthogonal Matrices," *Journal of Mathematics and Physics*, Vol. 12, 1933, pp. 311 - 320.
5. Plotkin, M., *Binary Codes with Specified Minimum Distance*, Moore School of Electrical Engineering, June 1952. (also *Transactions of the IRE*, Vol. IT-6, No. 4, September 1960.)
6. Hall, M., Jr., *A Survey of Difference Sets*, Proceedings of the American Mathematical Society, Vol. 7, 1956, pp. 975 - 986.
7. Brauer, A., "On a New Class of Hadamard Determinants," *Mathematische Zeitschrift*, Vol. 58, 1953, pp. 219 - 225.
8. Skolen, Th., Chowla, S., and Lewis, D. J., "The Diophantine Equation  $2^{n+2} - 7 = x^2$  and Related Problems," *Proceedings of the American Mathematical Society*, Vol. 10, 1959, pp. 663 - 669.
9. Todd, J. S., "A Combinatorial Problem," *Journal of Mathematics and Physics*, Vol. 12, 1933, pp. 321 - 333.
10. Williamson, J., "Hadamard's Determinant Theorem and the Sum of 4 Squares," *Duke Journal of Mathematics*, Vol. 11, 1944, pp. 65 - 81.
11. Bellman, R., *Introduction to Matrix Analysis*, McGraw-Hill, New York, 1960, p. 127.
12. Bose, R. C., "On the Construction of Balanced Incomplete Block Designs," *Annals of Eugenics*, Vol. 9, 1939, pp. 353 - 399.
13. Hall, M., Jr., "A Survey of Combinatorial Analysis" in *Some Aspects of Analysis and Probability* (by Kaplansky, Hall, Hewitt, and Fortet), John Wiley and Sons, 1958.
14. Hall, M., Jr., *Projective Planes and Related Topics*, Lectures at California Institute of Technology, 1954 (distributed by bookstore, CIT).
15. Stanton, R. G., and Sprott, D. A., "A Family of Difference Sets," *Canadian Journal of Mathematics*, Vol. 10, 1958, pp. 73 - 77.

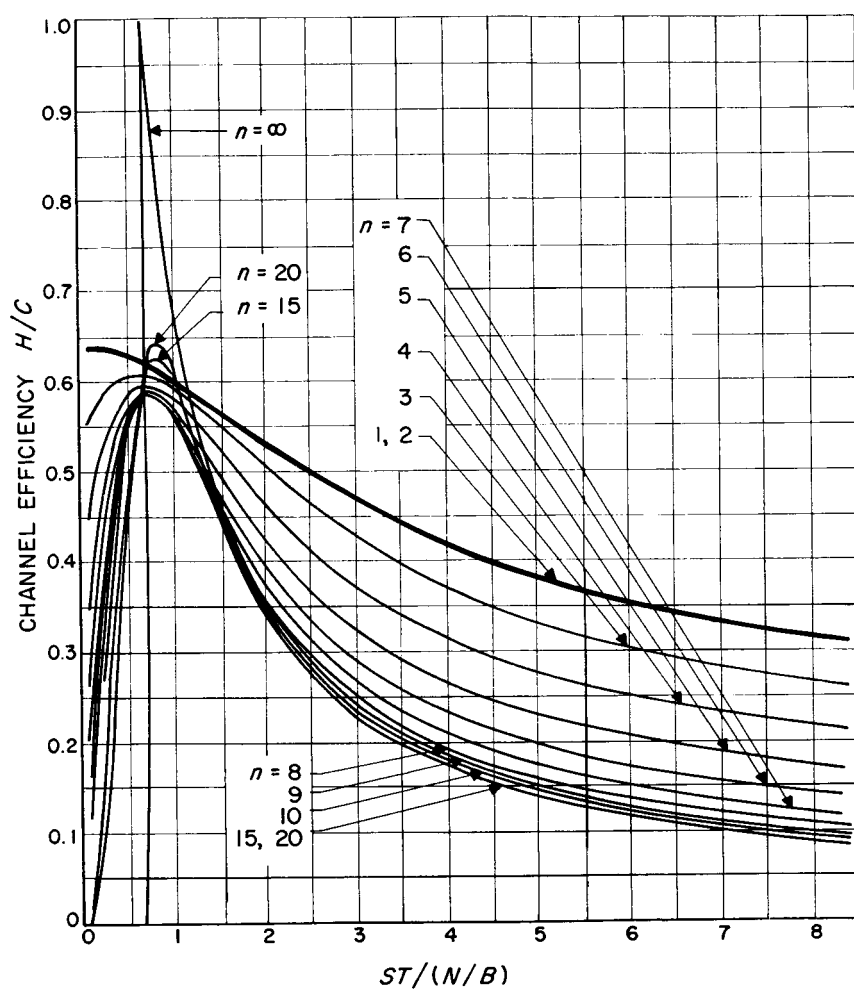


Fig. 40. Channel efficiency - biorthogonal codes

## REFERENCES (Cont'd)

16. Golomb, S. W., Welch, L. R., and Hales, A., *On the Factorization of Trinomials over GF(2)*, Memorandum No. 20-189, Jet Propulsion Laboratory, Pasadena, Calif., July 12, 1959.
17. Marsh, R. W., *Table of Irreducible Polynomials over GF(2) Through Degree 19*, distributed by Office of Technical Services, Commerce Dept., Washington, D. C., October 24, 1957.
18. Golomb, S. W., Gordon, B., and Welch, L. R., "Comma-Free Codes," *Canadian Journal of Mathematics*, Vol. 10, 1958, pp. 202-209.
19. Golomb, S. W., Welch, L. R., and Goldstein, R. M., *Cycles from Nonlinear Shift Registers*, Progress Report No. 20-389, Jet Propulsion Laboratory, Pasadena, Calif., August 31, 1959.
20. "The Generation of Periodic Pulse Rates," *Research Summary No. 4*, (June 1, 1959 to August 1, 1959), Jet Propulsion Laboratory, Pasadena, Calif., August 15, 1959 (Confidential).
21. Helstrom, C. W., "The Resolution of Signals in White Gaussian Noise," *Proceedings of the IRE*, Vol. 43, No. 9, September 1955, p. 1111.
22. Lawton, J. G., "Comparison of Binary Data Transmission Systems," *Second National Convention on Military Electronics, IRE*, 1958, p. 54.
23. Jaffe, R. M., and Rechten, E., "Design and Performance of Phase-Lock Circuits Capable of Near-Optimum Performance over a Wide Range of Input Signal and Noise Levels," *Transactions of the IRE, IT-1*, March 1955, pp. 66-76.
24. Woodward, P. M., and Davies, I. L., "Information Theory and Inverse Probability in Telecommunication," *Proceedings of the IEE*, Vol. 99, Part III, 1952, p. 37.
25. Davies, I. L., "On Determining the Presence of Signals in Noise," *Proceedings of the IEE*, Vol. 99, Part III, 1952, p. 45.
26. Fano, R. M., "Communication in the Presence of Additive Gaussian Noise," *Communication Theory*, W. Jackson, Ed., New York, Academic Press, Inc., 1953, pp. 169-182.
27. Middleton, D., and Van Meter, D., "Detection and Extraction of Signals in Noise From the Point of View of Statistical Decision Theory," *Journal of the Society of Industrial and Applied Mathematics*, Vol. 3, Part I, December 1955, pp. 192-253; Vol. 4, Part II, June 1956, p. 86.
28. Reed, I. S., "A Class of Multiple-Error-Correcting Codes and the Decoding Scheme," *IRE Transactions on Information Theory, PGIT-4*, September 1954, p. 38.
29. Green, J. H., Jr., and San Soucie, R. L., "An Error-Correcting Encoder and Decoder of High Efficiency," *Proceedings of the IRE*, Vol. 46, No. 10, October 1958, pp. 1741-1744.

## REFERENCES (Cont'd)

30. Birkhoff, G., and MacLane, S., *A Survey of Modern Algebra*, New York, Macmillan Co., 1953.
31. Shannon, C. E., "The Mathematical Theory of Communication," *Bell Systems Technical Journal*, Vol. 27 (July-October 1948), pp. 379 - 423, 623 - 56.